

TREBALL FI DE GRAU

**Grau en Enginyeria Electrònica Industrial i Automàtica.**

**DISSENY DEL SISTEMA DE CONTROL D'UN VEHICLE  
AUTÒNOM SUBMARÍ**



**Memòria i Annexos**

**Autor:** Gerard Bonavida Vila  
**Director:** Ivan Masmitjà i Rusinyol  
**Convocatòria:** Gener 2018



## **RESUM DEL PROJECTE**

En les últimes dècades l'àrea de l'electrònica avança feroçment en el mercat, oferint cada cop més possibilitats de computació i emmagatzematge de dades. Aquesta àrea està dedicada a satisfer les demandes tecnològiques tant d'empreses com usuaris.

Aquest fet ofereix la possibilitat de crear nous robots i vehicles autònoms, dotant-los fins i tot d'una intel·ligència artificial.

També es complementa amb la necessitat innata de l'ésser humà per investigar noves àrees de l'electrònica així com investigar els llocs inexplorats, inhòspits i hostils, on no arriba físicament una persona.

Aquest treball se centra en la necessitat d'una renovació de l'electrònica del Guanay II, un submarí autònom creat pel grup de recerca SARTI de la UPC.

La renovació i actualització dels components electrònics permet una major velocitat de computació i la possibilitat d'afegir noves eines que permeten una major robustesa a la unitat de control.

Així doncs, el treball es distribueix en diferents apartats que s'anomenen a continuació.

En primer lloc, es fan unes anàlisis del mercat actual per evaluar els diferents dispositius i sistemes que existeixen actualment, i finalment, veure cap on avança el mercat.

En segon lloc, es desenvolupa una adaptació del miniordinador NUC5i7RYH de la marca Intel, com a unitat de control del Guanay II.

En tercer lloc, es dissenya una nova arquitectura de l'electrònica de control del Guanay II, passant a anomenar-se Guanay 3.0. L'explicació d'aquest apartat és deguda a un canvi total de les tecnologies i la capacitat actual de més control i computació de dades del submarí.

Per finalitzar, s'investiga, desenvolupa i innova sobre un sistema d'encesa, necessària per al Guanay 3.0. Per fer-ho es dissenyen dos prototips, el bàsic i el final. Amdós dispositius són dissenyats des de zero i contenen un microcontrolador ATmega328P-PU. També s'estableix una connexió inalàmbrica a través del ordinador i del dispositiu dissenyat. En aquest apartat es passa d'un disseny fet amb ordinador a una placa física (PCB) on van soldats tots els components.

Les proves s'han dut a terme al laboratori i han permès verificar el seu correcte funcionament. Així doncs, s'han assolit els objectius inicials del projecte i també ha permès adquirir un coneixement més ampli en el camp de l'electrònica i dels AUV.

Els apartats han estat resolts satisfactòriament.

## **RESUMEN DEL PROYECTO**

En las últimas décadas el área de la electrónica avanza ferozmente en el mercado, ofreciendo cada vez más posibilidades de computación i almacenamiento de datos. Ésta área esta dedicada a satisfacer las demandas tecnológicas tanto de empresas como de usuarios.

Éste hecho ofrece la posibilidad de crear nuevos robots i vehiculos autónomos, dotándolos incluso de una inteligencia artificial.

También se complementa con la necesidad innata del ser humano para investigar nuevas áreas de la electrónica, así como investigar lugares inexplorados, inhospitos i hostiles, donde no llega físicamente una persona.

Éste trabajo se centra en la necesidad de una renovación de al electrónica del Guanay II, un submarino autónomo creado por el grupo de investigación SARTI de la UPC.

La renovación y actualización de los componentes electrónicos permiten una mayor velocidad de computación i la posibilidad de añadir nuevas herramientas que permiten una mayor robustez a la unidad de control.

Así pues, el trabajo se distribuye en diferentes apartados que se nombran a continuación.

En primer lugar, se hacen unos análisis del mercado actual para evaluar los diferentes dispositivos y sistemas que existen actualmente, i finalmente, ver hacia dónde avanza el mercado.

En segundo lugar, se desarrolla una adaptación del miniordenador NUC5i7RYH de la marca Intel, como unidad de control del Guanay II.

En tercer lugar, se diseña una nueva arquitectura de la electrónica de control del Guanay II, pasando a nombrarse Guanay 3.0. La explicación de éste apartado es debida a un cambio total de las tecnologías i la capacidad actual de más control i computación de datos del submarino.

Para finalizar, se investiga, desarrolla i innova sobre un sistema de encendido, necesáreo para el Guanay 3.0. Para eso, se diseñan dos prototipos, el básico i el final. Los dos dispositivos estan diseñados desde cero y contienen un microcontrolador ATmega328P-PU. También se establece una conexión inalámbrica a través del ordenador i del dispositivo diseñado. En este apartado se passa de un diseño hecho con ordenador a una placa física (PCB) donde van soldados todos los componentes.

Las pruebas se han llevado a cabo en el laboratorio y han permitido verificar su correcto funcionamiento. Así pues, se han alcanzado los objetivos iniciales del proyecto y también han permitido adquirir un conocimiento más amplio en el campo de la electrónica y de los AUV.

Los apartados han estado resueltos satisfactoriamente.

## **ABSTRACT**

In the last decades, the area of electronics is advancing fiercely in the market, offering more and more possibilities of computing and data storage. This area is dedicated to satisfying the technological demands of both companies and users.

This fact offers the possibility of creating new robots and autonomous vehicles, endowing them with an artificial intelligence.

It is also complemented by the innate need of the human being to investigate new areas of electronics, as well as to investigate unexplored, inhospitable and hostile places where a person does not physically arrive.

This work focuses on the need for a renewal of the electronics of Guanay II, an autonomous submarine created by the SARTI research group of the UPC.

The renewal and updating of the electronic components allow greater computing velocity and the possibility of adding new tools that allow a greater robustness to the control unit.

Thus, the work is distributed in different sections that are named below.

In the first place, some analyzes of the current market are made to evaluate the different devices and systems that currently exist, and finally, to see where the market advances.

Secondly, an adaptation of the NUC5i7RYH minicomputer of the Intel brand is developed, as the Guanay II control unit.

In third place, a new architecture of the Guanay II electronics control is designed, changing its name to Guanay 3.0. The explanation of this section is due to a total change of technologies and the current capacity of more control and data computation of the submarine.

Finally, we investigate, develop and innovate on a boot system, necessary for the Guanay 3.0. For that, two prototypes are designed, the basic and the final. The two devices are designed from scratch and contain an ATmega328P-PU microcontroller. A wireless connection is also established through the computer and the prototype device. In this section we go from a design made with a computer to a physical board (PCB) where all the components are welded.

The tests have been carried out in the laboratory and have allowed to verify its correct functioning.

Thus, the initial objectives of the project have been achieved and they have also allowed to acquire a broader knowledge in the field of electronics and AUV.

The sections have been resolved satisfactorily.





## **AGRAÏMENTS**

Després d'un intens període de temps, puc escriure amb cert respir aquests agraïments.

Ha estat un aprenentatge intens i dur a nivell tècnic però també a nivell personal.

Ara és el moment de donar el meu més sentit agraïment a totes aquelles persones que m'han ajudat en aquest camí.

Primer de tot, dono les gràcies per haver-me facilitat un espai a la Universitat per poder desenvolupar el treball amb total llibertat.

A continuació, vull agrair la col·laboració del director d'aquest TFG, Ivan Masmitjà i del professor Spartacus Gomariz. Gràcies pels vostres comentaris i consells, m'han ajudat a afinar la part tècnica.

També vull agrair la paciència dels meus pares, la meva germana, i familiars que han tingut amb mi tots aquests anys.

Per últim, agrair a tots els amics i companys que m'han animat en els moments que més ho necessitava.

De veritat, moltes gràcies a tots.

Gerard Bonavida Vila



## ÍNDEX

<b>RESUM DEL TREBALL</b>	<b>II</b>
<b>RESUMEN DEL TRABAJO</b>	<b>IV</b>
<b>ABSTRACT</b>	<b>VI</b>
<b>AGRAÏMENTS</b>	<b>8</b>
<b>CAPÍTOL 1. INTRODUCCIÓ</b>	<b>20</b>
1.1. Estat de l'art	20
1.1.1. Vehicles autònoms submarins	21
1.2. Guanay II	25
1.2.1. Funcionament general	26
1.3. Unitat de control	27
1.4. Electrònica de la unitat de control	28
1.5. Software de la unitat de control	29
1.6. Objectius	29
<b>CAPÍTOL 2. ANÀLISIS DEL MERCAT</b>	<b>30</b>
2.1. Introducció	30
2.2. Anàlisi Software	30
2.2.1. SO per microprocessadors	32
2.2.2. SO per microcontroladors	36
2.3. Anàlisi de Hardware	37
2.3.1. FPGA vs Microcontrolador vs Microprocessador	37

2.3.2. Sistemes “Embedded” basats amb ARM .....	44
2.3.3. Miniordinadors .....	49
2.3.4. Microcontroladors Atmega328 vs MC9S08PT32 .....	51
2.4. Futur del mercat .....	53
<b>CAPÍTOL 3. GUANAY 2.1 .....</b>	<b>56</b>
3.1. PC104+ .....	56
3.2. Unitat de control del Guanay II i Guanay 2.1 .....	59
3.2.1. Estructura de control del Guanay II .....	59
3.2.2. Estructura de control del Guanay 2.1 .....	61
3.3. Proves experimentals .....	62
3.3.1. Sensor de temperatura LM35 .....	62
3.3.2. Industrial Pressure Sensor .....	63
3.3.3. Introducció de la implementació amb Labview .....	64
3.3.4. Anàlisi detallat del programa .....	67
3.4. Conclusions .....	72
<b>CAPÍTOL 4. GUANYAY 3.0 .....</b>	<b>73</b>
4.1. Introducció .....	73
4.2. Objectius .....	73
4.3. Nova arquitectura Unitat de Control .....	74
4.3.1. Guanay II .....	74
4.3.2. Guanay 3.0 .....	75
4.4. Funcionament de la nova arquitectura .....	81

4.5. Consum potencia .....	82
4.6. Justificació de la nova arquitectura .....	83
4.7. Conclusions .....	84
4.8. Nous horitzons .....	85
<b>CAPÍTOL 5. DISSENY DEL SISTEMA D'ENCESA PER AL GUANAY 3.0 .....</b>	<b>86</b>
5.1. Prototip I. Circuit bàsic amb el microcontrolador ATmega328P .....	86
5.1.1. Altres problemes .....	91
5.2. Prototip II. Disseny final del circuit d'encesa .....	92
5.2.1. Descripció de pins .....	93
5.2.2. Control manual i automàtic .....	95
5.2.3. Sistema d'encesa software .....	95
5.2.4. Comandes registrades .....	97
5.2.5. Xbee .....	98
5.2.6. Esquema placa general .....	102
5.2.7. Problemes del prototip II .....	106
5.3. Resultat final del prototip I i prototip II .....	111
5.4. Conclusions .....	111
<b>CONCLUSIONS FINALS .....</b>	<b>113</b>
<b>PRESSUPOSTOS .....</b>	<b>115</b>
<b>BIBLIOGRAFIA .....</b>	<b>126</b>
<b>ANNEX I .....</b>	<b>134</b>
<b>ANNEX II .....</b>	<b>142</b>

## ÍNDIX DE FIGURES I TAULES

### CAPÍTOL 1. INTRODUCCIÓ

Figura 1.0. Esquema general de vehicles autònoms .....	20
Figura 1.1. Esquema de vehicles autònoms marins .....	21
Figura 1.2. Submarí tripulat .....	21
Figura 1.3. Submarí no tripulat .....	21
Figura 1.4. ROV .....	22
Figura 1.5. AUV .....	22
Figura 1.6. AUV Propulsat .....	22
Figura 1.7. AUV Glider .....	23
Figura 1.8. Moviment AUV Glider .....	23
Figura 1.9. AUV Biomimètic .....	23
Figura 1.10. Estructura AUV WaveGlider .....	24
Figura 1.11. Moviment AUV WaveGlider .....	24
Figura 1.12. Guanay II .....	25
Figura 1.13. Funcionament del Guanay II .....	26
Figura 1.14. Estructura de funcionament del Guanay II .....	26
Figura 1.15. Placa PC104+ .....	27
Figura 1.16. Interior Guanay II .....	28

### CAPÍTOL 2. MERCAT

Figura 2.0. Diagrama blocs funcionament dispositiu .....	31
Figura 2.1. "Logos" de diferents sistemes operatius .....	31

Figura 2.2. Rendiment SO Windows 10 (esquerra) i distribució de tasques (dreta) .....	32
Figura 2.3. FPGA ice40LPKQN84 de Lattice (esquerra) i FPGA spartan XC3S250AN de Xilinx (dreta) .....	38
Figura 2.4. Estructura interna d'una FPGA .....	38
Figura 2.5. Microcontrolador ATmega328P de Microchip (esquerra) i microcontrolador TMS320F28335 de Texas Instrument (dreta) .....	39
Figura 2.6. Arquitectura del microcontrolador ATmega328p .....	40
Figura 2.7. Arquitectura del microcontrolador LM4F120 .....	41
Figura 2.8. Posicions dels diferents sistemes segons la computació de dades, preu i consum elèctric .....	41
Figura 2.9. Gràfic de proveïdors de microcontroladors a nivell mundial el 2015 .....	42
Figura 2.10. Arquitectura de Von Neumann .....	43
Figura 2.11. Raspberry Pi 2 Model B (esquerra) i Raspberry Pi 3 Model B (dreta) .....	44
Figura 2.12. Disposició dels pins de la Raspberry Pi 3 B .....	46
Figura 2.13. BeagleBone Black .....	47
Figura 2.14. Pins de la BeagleBone Black .....	48
Figura 2.15. Intel NUC (esquerra) i Asus VivoMini (dreta) .....	49
Figura 2.16. Microcontrolador ATmega328P .....	51
Figura 2.17. Microcontrolador LQFP32 .....	52
Figura 2.18. Diagrama de blocs del SoC de la marca Nordic Semiconductor .....	54
Figura 2.19. Gràfic del mercat del IoT .....	55
Taula 2.0. SO de Windows més rellevants ordenats cronològicament .....	33
Taula 2.1. Microcontroladors que admeten el SO FreeRTOS .....	37

Taula 2.2. Característiques de la Raspberry Pi 2 B i Raspberry Pi 3 B .....	45
Taula 2.3. Característiques de la BeagleBone Black .....	48
Taula 2.4. Característiques de Intel NUC5iRYH i de Asus VivoMini UN65 .....	50
Taula 2.5. Venta de microcontroladors (en milions de dòlars) .....	51
Taula 2.6. Característiques del ATmega328P i del MC9S08PT32 .....	52

### **CAPÍTOL 3. Guanay 2.1**

Figura 3.0. PC104 (esquerra) i Intel NUC5i7RYH (dreta) .....	56
Figura 3.1. Arquitectura Guanay II .....	59
Figura 3.2. HIRK-315AP .....	60
Figura 3.3. CTD XR-420 .....	61
Figura 3.4. Arquitectura Guanay 2.1 .....	61
Figura 3.5. Sensor LM35 .....	63
Figura 3.6. Sensor de pressió .....	63
Figura 3.7. DAQ USB-6008 .....	64
Figura 3.8. Panell frontal Labview. Programa parat .....	65
Figura 3.9. Diagrama blocs Labview. Programa parat .....	66
Figura 3.10. Panell Frontal (esquerra) i diagrama de blocs (dreta). Programa en funcionament .....	66
Figura 3.11. Diagrama blocs DAQ Assistant .....	67
Figura 3.12. Diagrama de blocs amb gràfic .....	68
Figura 3.13. Lectura valors temperatura i pressio en volts .....	68
Figura 3.14. Diagrama blocs temperatura .....	68
Figura 3.15. Valor de la temperatura amb graus .....	69



Figura 3.16. Diagrama blocs sensor pressió .....	70
Figura 3.17. Valors de pressió en Bar (sobre) i hPa (sota) .....	70
Figura 3.18. DAQ amb sensor de temperatura i pressió .....	71
Taula 3.0. Comparativa entre PC104 i Intel NUC5i7RYH .....	57

## **CAPÍTOL 4. Guanay 3.0**

Figura 4.0. Arquitectura Guanay II .....	74
Figura 4.1. Esquema bàsic del Guanay 3.0 .....	75
Figura 4.2. Arquitectura del Guanay 3.0 .....	76
Figura 4.3. Dispositiu Intel NUC .....	77
Figura 4.4. Dispositiu BeagleBone Black .....	77
Figura 4.5. Safety: Raspberry Pi 3 Model B .....	78
Figura 4.6. Funcionament del sistema de seguretat .....	79
Figura 4.7. Boot System: ATmega328P-PU .....	80
Figura 4.8. Bus I2C .....	84
Taula 4.0. Potència Guanay II .....	82
Taula 4.1. Potència Guanay 3.0 .....	83

## **CAPÍTOL 5. Disseny del sistema d'encesa per al Guanay 3.0**

Figura 5.0. Esquema pel funcionament bàsic del prototip I .....	87
Figura 5.1. Prototip II en protoboard .....	87
Figura 5.2. USB a TTL model FT232RL .....	88
Figura 5.3. Selecció microcontrolador i port (COM) .....	89
Figura 5.4. Esquema ATmega328P-PU amb polsador .....	90

Figura 5.5. Compilació programa .....	90
Figura 5.6. Pujada programa .....	91
Figura 5.7. Programa pujat .....	91
Figura 5.8. Circuit enclavament Raspberry .....	93
Figura 5.9. Diagrama d'estats del sistema d'encesa .....	96
Figura 5.10. Xbee S2C .....	98
Figura 5.11. Xarxa amb forma d'estrella .....	98
Figura 5.12. XCTU amb els dos Xbee .....	99
Figura 5.13. Mòdul A000021 per adaptar el Xbee a Arduino Uno .....	99
Figura 5.14. Esquema per al funcionament de la comunicació RF .....	100
Figura 5.15. Comunicació a través del xat de XCTU. Missatge enviat (vermell) i missatge rebut (blau) .....	100
Figura 5.16. Pins Xbee S2C .....	101
Figura 5.17. Conexió del ATmega328P-Pu amb els pins referenciats .....	102
Figura 5.18. Power system .....	102
Figura 5.19. Sistema de relés i interruptors (manual/automàtic) .....	104
Figura 5.20. Leds indicadors d'estat. Groc, vermell i verd .....	105
Figura 5.21. PCB amb la distribució dels components del prototip II .....	105
Figura 5.22. Mòdul A000021 amb Xbee, ATmega328P-PU i Arduino Uno .....	107
Figura 5.23. Pics de voltatge prototip II. V: 1 V/DIV, H: 500 ns/DIV .....	108
Figura 5.24. Solució pels rebots del polsador .....	108
Figura 5.25. Solució pels pics de voltatge .....	109
Figura 5.26. Interruptor erroni (esquerra) i interruptor correcte (dreta) .....	109

Figura 5.27. Situació de nous components al prototip II .....	110
Figura 5.28. “Bottom” de la PCB .....	110
Figura 5.29. Prototip I .....	111
Figura 5.30. Prototip II amb el Xbee .....	111
Taula 5.0. Pins del prototip II. Disseny final del sistema d’encesa .....	95
Taula 5.1. Pins requerits per al funcionament del Xbee .....	101



## **CAPÍTOL 1. INTRODUCCIÓ**

En el primer capítol hi ha una introducció del treball de final de grau d'Enginyeria Electrònica Industrial i Automàtica.

El primer apartat es parla sobre l'estat de l'art dels vehicles autònoms i centra l'atenció amb els vehicles autònoms submarins.

En segon lloc s'introdueix al lector al Guanay II, un vehicle submarí autònom dissenyat per la UPC.

Finalment, s'exposen els objectius que té el treball.

### **1.1. Estat de l'art**

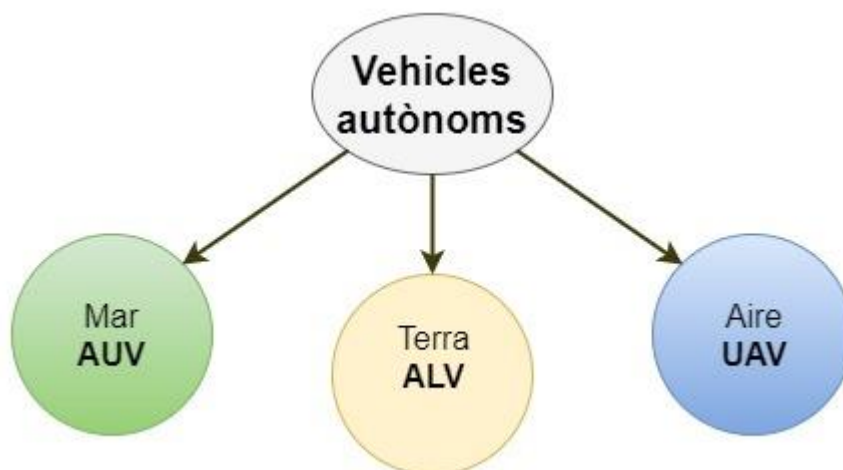
En els últims anys els fabricants de vehicles estan centrant la seva atenció en els vehicles autònoms. Això és degut a que la tecnologia necessària per fer funcionar un vehicle autònom d'aquestes característiques ja pot ser integrada dins del vehicle.

En el sector del automòbil està sent una gran revolució degut a la necessitat de renovació, innovació i millora dels vehicles.

Les avantatges principals que ens aporten aquests vehicles són:

- Van lligats a una tecnologia més neta on els motors per fer-los moure són elèctrics.
- Un vehicle autònom no requereix de conductor. Molt útil per fer els vehicles més reduïts, econòmics i destinats a tasques perilloses.
- Eliminació del error humà.

El seu ús és cada cop més extens i estan en tots els camps: mar, terra i aire.



*Figura 1.0. Esquema general de vehicles autònoms*

### 1.1.1. Vehicles autònoms submarins

Els vehicles autònoms submarins, amb anglès Autonomous Underwater Vehicle (AUV), van sorgir degut a la necessitat d'una exploració més detallada del fons marí, així com de les bases petrolíferes i detecció de mines marines entre d'altres. Arribant a llocs on inabastables per l'home.

Aquests vehicles permeten treballar amb més altes pressions i sense limitacions de llum.

Aquest tipus de vehicles es poden classificar com a tripulats i no tripulats.

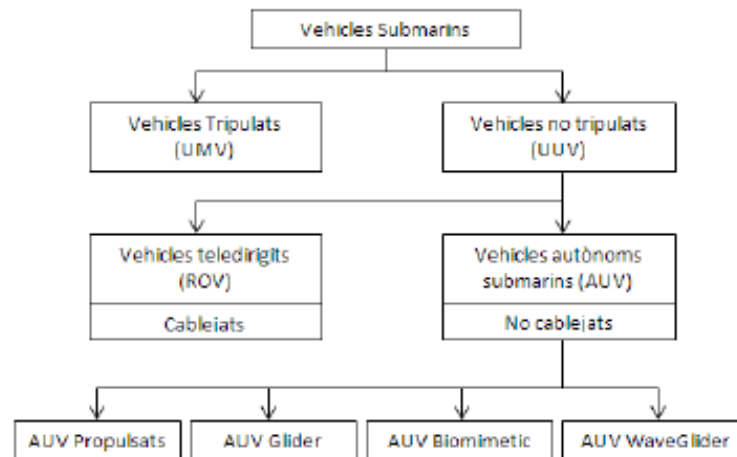


Figura 1.1. Esquema de vehicles autònoms marins [1].

L'esquema de la Figura 1.1 està distribuït de la següent manera.

Els vehicles submarins estan classificats en dos grans blocs: els tripulats (UMV) i els no tripulats (UUV). Els primers tindran un o més operaris dins el propi submarí.



Figura 1.2. Submarí tripulat [2].



Figura 1.3. Submarí no tripulat [3].

En el bloc dels vehicles no tripulats (UUV) s'hi troben els vehicles teledirigits (ROV) i els vehicles autònoms (AUV). La diferència que hi ha entre els dos, és que els ROV són vehicles submarins cablejats on l'operari està a la superfície dirigint el submarí, en canvi, els AUV són vehicles autònoms, no necessiten cap operari dirigint el vehicle, sinó que ell sol fa les tasques que té encomanades, prèviament carregades al la unitat de control pels operaris.



*Figura 1.4. ROV [4].*



*Figura 1.5. AUV [5]*

Dins dels vehicles (AUV) hi trobem quatre grups principals:

- AUV Propulsats:  
Vehicles autònoms propulsats per motors a propulsió o amb hèlices. Solen ser els més habituals.



*Figura 1.6. AUV Propulsat [6].*

- AUV Glider:  
També anomenats planadors, es propulsen amb unes ales, mitjançant un moviment degut a un canvi de flotabilitat.



Figura 1.7. AUV Glider [7].

El seu funcionament permet convertir el moviment vertical en horitzontal. Així, s'obté un sistema de molt baix consum elèctric.

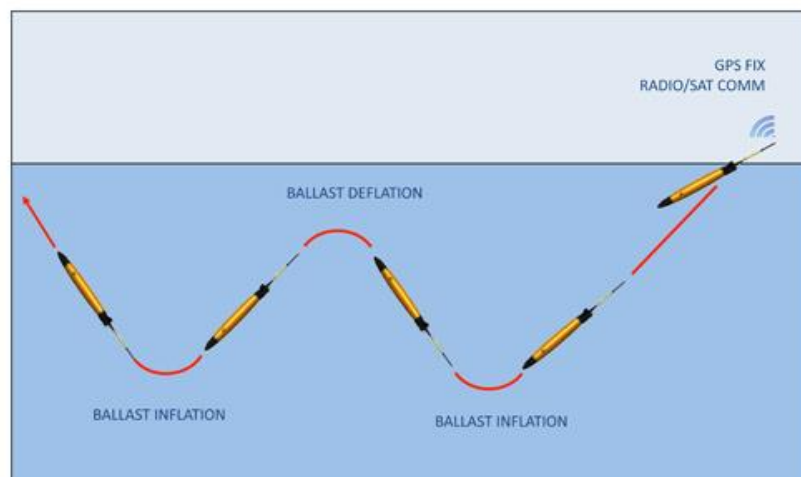


Figura 1.8. Moviment AUV Glider [8].

AUV Biomimètics [9]:

Presenten una estructura que emula els moviments dels animals marins. Els més habituals són els que tenen una estructura amb forma de peix, tot i que també hi ha en forma de tortuga entre d'altres.



Figura 1.9. AUV Biomimètic [10]



- AUV WaveGlider:  
També anomenats patins, són vehicles que es mouen mitjançant la força de les onades.  
Tenen un molt baix consum elèctric ja que inclouen una placa solar a la superfície per autoregenerar les bateries. Aquestes només gestionen el control del vehicle, no per la propulsió del mateix.

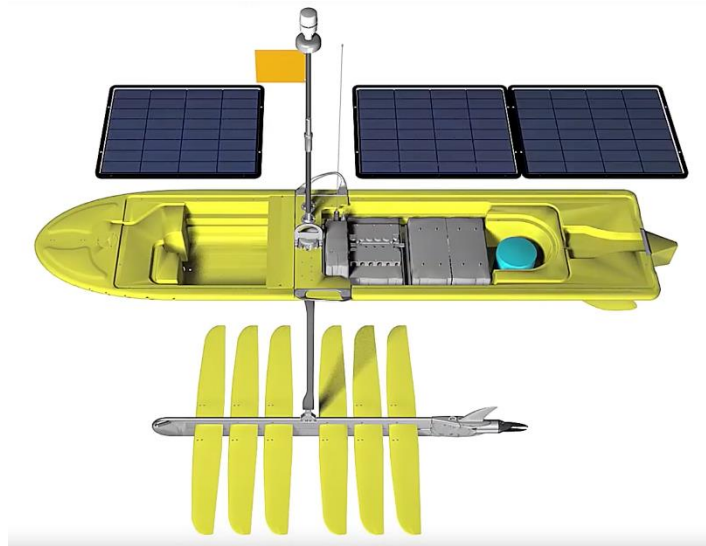


Figura 1.10. Estructura AUV WaveGlider [11].

En Figura 1.11 es pot veure el funcionament del vehicle.

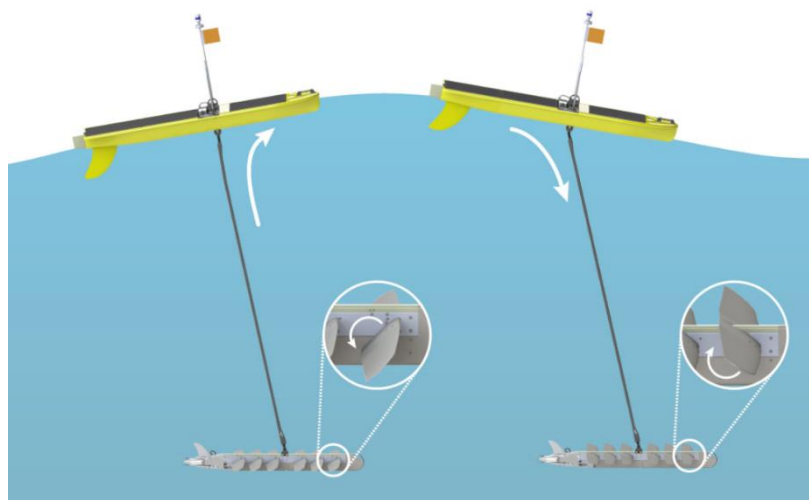
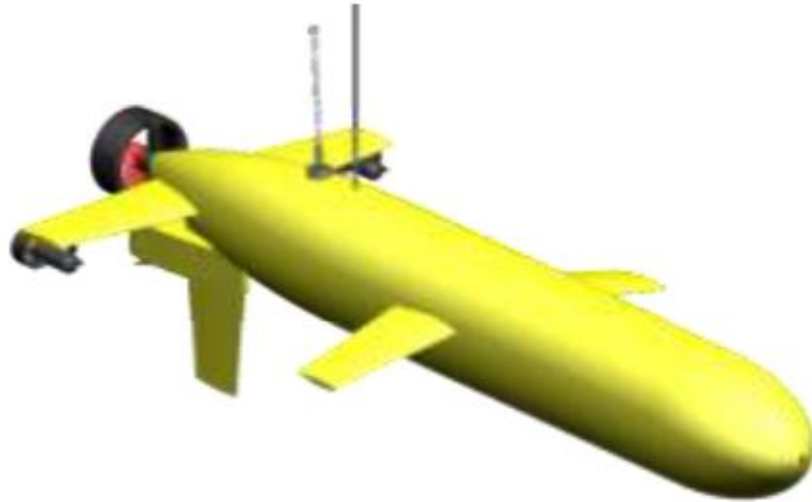


Figura 1.11. Moviment AUV WaveGlider.

## 1.2. Guanay II

El Guanay II (Figura 1.12) és un AUV dissenyat per la UPC, en concret pel grup de recerca de Sistemes d'Adquisició Remota i Tractament de la Informació (SARTI).



*Figura 1.12. Guanay II [1].*

És un vehicle autònom dirigit i propulsat per hèlices, en concret, una hèlix per la propulsió i dues pel moviment de direcció.

El seu disseny està pensat per la investigació oceanogràfica.

La seva funció dins de la investigació oceanogràfica se centra en realitzar mesures en columnes d'aigua. Les columnes d'aigua són perfils verticals que hi ha dins d'una superfície marina en un punt concret.

El moviment vertical s'aconsegueix mitjançant un canvi de flotabilitat del vehicle, que provoca poques perturbacions en el medi.

En la figura següent es pot veure el funcionament del Guanay II.

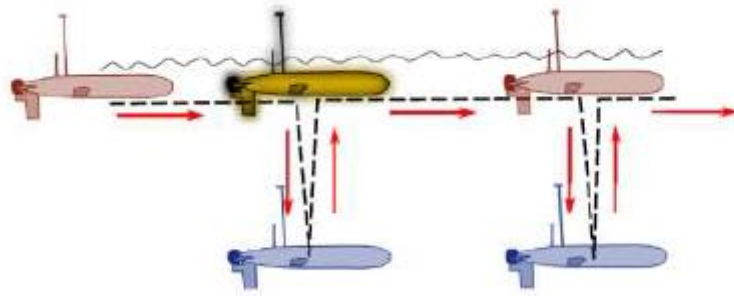


Figura 1.13. Funcionament del Guanay II [2].

### 1.2.1. Funcionament general

Per al funcionament del Guanay II són necessaris dos equips bàsics, un a l'exterior per rebre i tractar les dades, i prendre decisions, i un altre a l'interior, propi del vehicle per poder desenvolupar les tasques autònomament.

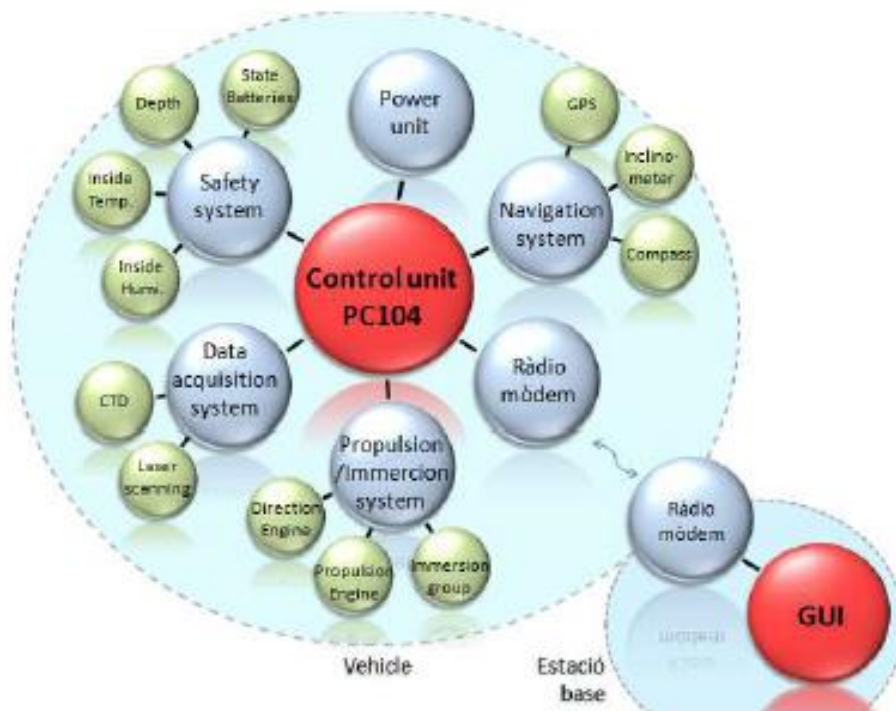


Figura 1.14. Estructura de funcionament del Guanay II [2]-

A continuació es detalla l'estructura de funcionament mostrat a la Figura 1.14.

Bloc 1. Interior del Guanay II:

- Sistema de control i tractament de dades (Control Unit PC104+).
- Sistema de seguretat (Safety System).
- Sistema d'energia (Power Unit).
- Sistema de navegació (Navigation System).
- Sistema de comunicació (Inclou Ràdio mòdem i Wifi)
- Sistema de propulsió i immersió (Propulsion/Immercion System)
- Sistema d'adquisició de dades (Data acquisition System)

Bloc 2. Estació base a l'exterior del Guanay II:

- Ordinador per al control (GUI) del vehicle.
- Sistema de comunicació (Inclou Radiomòdem i Wifi)

El llenguatge de programació amb el que ha estat programada la unitat de control PC104+ és el Labview.

A l'estació base hi trobarem una interfície gràfica que ens mostrarà la ruta del vehicle, i dades com l'estat del submarí entre d'altres.

### 1.3. Unitat de control

La unitat de control principal del Guanay II és una placa "Embedded" anomenada PC104+.

El PC104+ és una placa de la companyia AMD que utilitza un processador Geocode LX800 a 500MHz.



*Figura 1.15. Placa PC104+ [1].*

Té un consum màxim d'uns 12 W. Disposa d'una ranura per posar una memòria DDR S0-DIMM, també conté una ranura per incloure una memòria compact Flash.

#### 1.4. Electrònica de la unitat de control

La unitat de control del Guanay II, que és el PC104+, és una placa que té més de 20 anys. És evident que l'electrònica actual amb la de fa 20 anys no té res a veure en quant a capacitat de tractament i emmagatzement de dades, així com el tamany de la placa.

Actualment, els sistemes "Embedded" (com el PC104+) són molt extensos en el mercat. Si comparem el PC104+ amb qualsevol dispositiu actual en quant a preu, les seves característiques de computació quadrupliquen, en el pitjor dels casos, les característiques del PC104+.

Un exemple de preus en quant a la computació de dades, és a dir, comparant dos processadors, el PC104+ que costa uns 400 € equivaldria a una Raspberry Pi 1 Model A que costa uns 25 €. Així doncs, es pot arribar a la conclusió de que l'electrònica de control de la que disposa el Guanay II està completament desfasada.

La disposició de tota la electrònica, inclòs el sistema de bateries, se situa en un cilindre completament estanc. En la següent figura es pot veure la seva disposició.

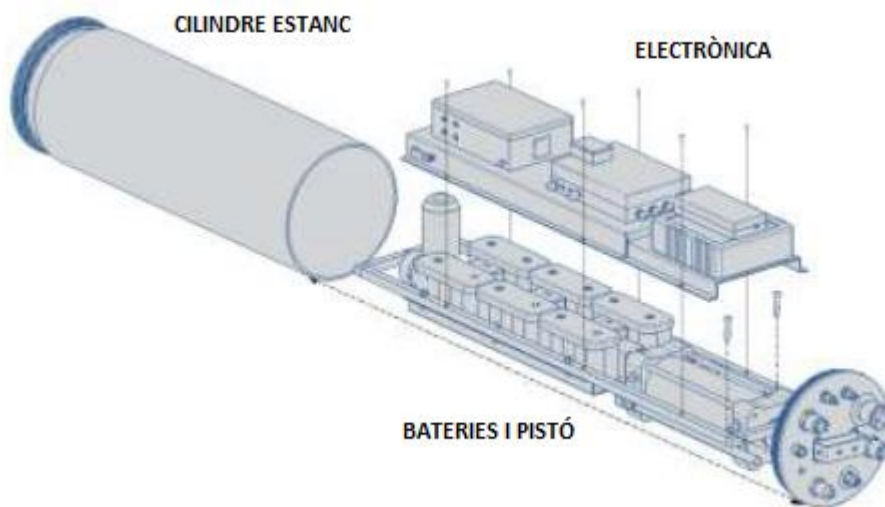


Figura 1.16. Interior Guanay II [1].

### **1.5. Software de la unitat de control**

El software (interfície) per fer funcionar la unitat de control és el Labview.

Labview és una plataforma de programació basada en objectes de pagament.

Actualment, hi ha multitud de plataformes (software) lliures per a la programació dels diferents dispositius. Aquest tipus de software s'anomena Open Source (codi obert).

El software lliure per a la programació de microcontroladors més famosa i actual és Arduino, i en quant a microprocessadors la plataforma més utilitzada és Linux o derivats d'aquest.

### **1.6. Objectius**

El treball està basat en quatre blocs:

Anàlisi de mercat, Guanay 2.1, Guanay 3.0 i I+D+I.

Cada capítol tindrà les seves pròpies conclusions ja que el treball de fi de grau es presenta com una anàlisi general sobre les unitats de control.

Per tant, els objectius comuns que es persegueixen en els capítols és la recerca veraç i eficaç de la informació, utilització de plataformes Open Source i investigació d'aquestes, resoldre problemes de seguretat en la unitats de control del Guanay II, la recerca de tecnologies actuals i per últim, la correcta planificació del treball.

També es vol dotar al Guanay d'una nova unitat de control, que permeti unes majors prestacions i noves funcionalitats al vehicle. Millorant la computació de dades, la robustesa de la unitat de control, la precisió durant la navegació i una major adquisició de dades entre d'altres. Ja que actualment, amb el PC104+ el sistema es satura.

En definitiva, es vol fer un estudi sobre una actualització de mitja vida de la unitat de control del Guanay II.

## **CAPÍTOL 2. ANÀLISIS DEL MERCAT**

### **2.1. Introducció**

En aquest apartat es fa una anàlisi sobre les diferents tecnologies que s'inclouen en aquest treball, centrat amb el control dels vehicles.

### **2.2. Anàlisi Software**

En aquest apartat es parla sobre el diferent software que existeix actualment.

S'ha decidit fer una anàlisi dels sistemes operatius per al control de vehicles, ja que la tecnologia avança molt ràpidament i en conseqüència, les facilitats per tenir-la, tant econòmicament com per treballar, estudiar, computar dades,...

Aquest estudi només es basa en les principals eines de software, ja que és impossible fer un anàlisi complet de tot el software que hi ha al mercat ja que és absolutament immensa.

En aquest estudi es fa especial èmfasi amb les idees que es van perseguint en tot el treball: Open Source, comunitat, expansibilitat i econòmic.

## Sistemes Operatius (SO)

Un sistema operatiu és el software principal o conjunt de programes d'un sistema informàtic que formen una capa de gestió entre la aplicació que volem desenvolupar i el hardware (components).

Aquest inclou un conjunt d'eines de sistema operatiu com entorns gràfics, explorador de fitxers, navegador web, programes predeterminats i d'altres.

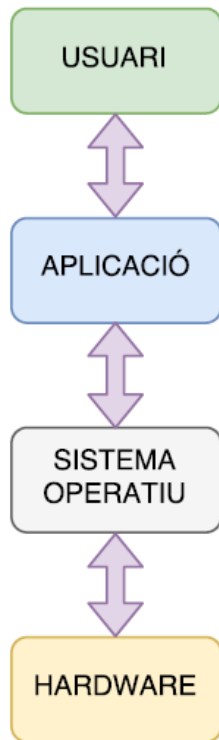


Figura 2.0. Diagrama blocs funcionament dispositiu.



Figura 2.1. "Logos" de diferents sistemes operatius [12].

Existeixen infinitat de sistemes operatius. Si ens movem en un apartat més informàtic trobarem sistemes operatius molt potents, per tractar gran quantitat de dades. Ara bé, per totes les aplicacions no necessitem uns sistemes operatius potents, com és el cas dels microcontroladors. Per fer un anàlisi més detallat s'explicarà dos tipus de sistemes operatius, els que s'utilitzen per informàtica (microprocessadors) i per control de sistemes més simples (microcontroladors).

Tots els SO són seqüencials, és a dir, no poden executar totes les tasques/processos alhora, sinó que ho fan un per un.

A nivell d'usuari no podem veure l'execució d'aquests processos degut a la seva rapidesa.



Actualment, aquesta velocitat en qualsevol SO per ordinador sol treballar a uns 3,0 GHz, que serien uns  $3,33 \cdot 10^{-10}$  segons. Per tant, veure seqüencialment aquestes tasques és impossible.

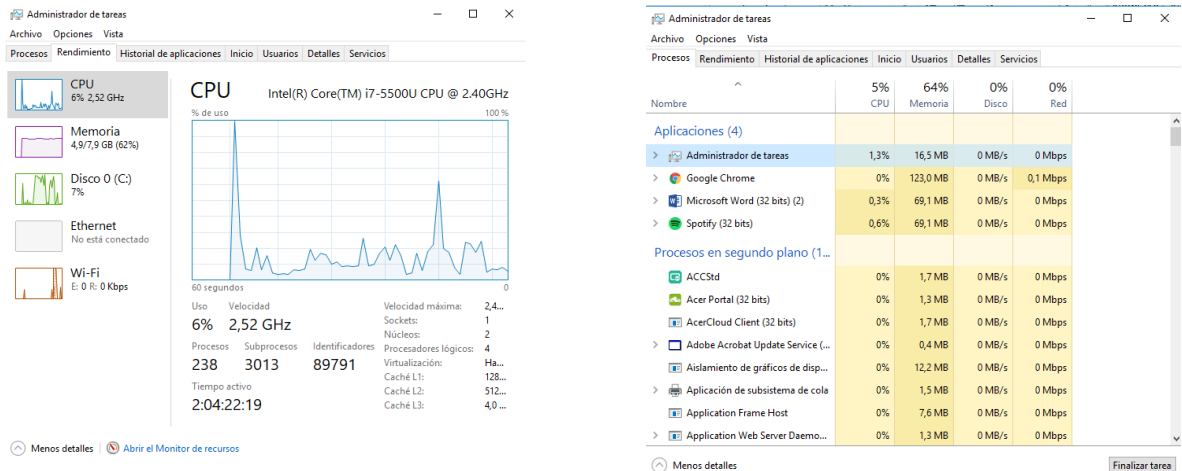


Figura 2.2. Rendiment SO Windows 10 (esquerra) i distribució de tasques (dreta)

## 2.2.1. SO per microprocessadors

### Windows 10

És un sistema operatiu desenvolupat per Windows [13]. És el SO més actual que té Windows i va sortir al mercat a mitjans de 2015.



Generalment s'utilitza en microprocessadors, que són els cervells dels ordinadors.

Windows és el sistema operatiu més utilitzat en tot el món, tant per tenir una interfície fàcil i amigable per tot usuari com per les fortes campanyes de màrketig que duen a terme.

El primer SO que va llançar Windows va ser l'any 1985 i s'anomenava Windows 1.0. Competia directament amb el SO d'Apple. No va tenir massa èxit degut a la poca funcionalitat que tenia.

Evidentment, Windows va seguir desenvolupant SO fins a dia d'avui.

La llista de SO és extensa, per tant, en la següent taula [posar número taula] s'anomenen els SO de Windows que van introduir un canvi notable al món informàtic:

Sistema Operatiu	Any
Windows 1.0	1985
Windows 3.0	1990
Windows 95	1995
Windows 2000	2000
Windows XP	2001
Windows Vista	2007
Windows 10	2015

Taula 2.0. SO de Windows més rellevants ordenats cronològicament.

Pel que fa a Windows 10, conté una gran comunitat, suport per part de l'empresa, interfície amigable per l'usuari, facilitat per trobar components, programes, drivers i d'altres.

Ara bé, té força inconvenients, i si bé no es noten a nivell d'usuari (que no li donarà un ús concret) sí que presenta inconvenients per aplicacions concretes o SO dedicats.

A continuació es detallen els problemes que es troben a l'utilitzar Windows 10 o qualsevol SO de Windows:

1) Actualitzacions constants

És una bona opció si no necessitem que el SO estigui completament dedicat a una aplicació.

Windows 10 permet ometre les actualitzacions però no és 100% funcional, és a dir, que hi haurà cops que se'ns actualitzarà igualment.

Per solucionar aquest problema s'haurà de "hackejar" el SO, que consisteix entre d'altres tasques en desactivar el SO del rellotge del PC.

2) Aplicacions inservibles

Per defecte, hi ha instal·lades unes aplicacions, que si bé com a usuari ens poden resultar útils, com a SO dedicat ens seran inservibles. Ocuparan memòria tant ROM com RAM, espai i processos.

3) Procesos inservibles

Windows 10 utilitza processos inservibles, que a nivell de SO dedicat ens faran perdre rapidesa en tractament de dades.

4) Numeració de processos

No es poden numerar els processos. Això és crític en un SO dedicat.

5) No apte per microcontroladors

Degut a la quantitat de recursos que utilitza i l'arquitectura del SO, és incompatible amb la majoria de microcontroladors que existeixen al mercat.

6) Virus

La gran majoria de virus que hi ha, estan digirits al SO de Windows, ja que és el SO més utilitzat en tot el món. Això suposa un gran problema per la integritat del propi sistema.

7) Preu

Windows 10 és un SO privatiu, i per tant, d'una empresa privada.

El seu preu oscila entre 50-2000 € segons la versió que es vulgui del SO i dels diferents serveis que es volen demanar a l'empresa.

8) No és una plataforma Open Source

No es poden veure ni obtenir les fonts del SO per poder canviar les diferents característiques d'aquest.

Per tant, s'arriba a la conclusió de que a nivell d'usuari és un bon sistema operatiu, amb una gran comunitat i serveis de la pròpia empresa per solucionar problemes.

A nivell de SO dedicat per a una aplicació concreta, com és la unitat de control del Guanay, resulta inconcebible tenir un SO que no es pot controlar al 100%.

### **Mac OS (Macintosh Operating System)**



És un sistema operatiu desenvolupat per Apple [14]. El seu primer SO anomenat “classic” Mac OS, va sortir al mercat l’any 1984 i anava dirigit al públic en general.

Comptava amb una interfície gràfica composta per un ratolí (mouse), finestres, icones i menús.

El SO més recent és el Mac OS i compta amb un gran nombre d’usuaris, ja que la interfície gràfica és molt amigable per l’usuari i li aporta una exclusivitat que altres sistemes operatius no donen.

A més, està altament protegit de virus, cosa que altres SO no poden fer gaire ostentació. Per suposat, també compta amb un gran suport per part de l’empresa Apple.

Mac OS té els mateixos inconvenients que Windows, fins i tot els incrementa (per exemple el preu, la exclusivitat i la falta de configuracions). No obstant això, els virus en Mac OS es redueixen notablement.

Per tant, és el SO menys ideal per un SO dedicat.

### **GNU/Linux**



És un sistema operatiu lliure gratuït basat amb la combinació de diversos projectes entre els quals es destaquen GNU i Free Software Foundation [15].

Tot el seu codi font pot ser obtingut, modificat i redistribuït gratuïtament subjecte a les normes de la GNU (General Public Licence).

Permet suportar plataformes com DEC Alpha, x86, AVR32, ARM d’entre altres.

Això vol dir que pot anar dedicat tant a microprocessadors (DEC Apha, x86, ARM) com a micro-controladors (AVR32, ARM).

Tot és configurable, per tant, podem tenir un control del 100% dels processos, de les actualitzacions i de les tasques que s’han de desenvolupar. Per un sistema dedicat ens serà de gran utilitat.

Aquest SO ha permès que existeixin derivats del propi sistema, per exemple Android.

Hi ha versions que tenen unes millors interfícies gràfiques per ser utilitzades per l'usuari i d'altres que tindran una interfície gràfica més pobre.

Al darrere d'aquest SO hi trobem una extensa comunitat, que permet que el software lliure segueixi avançant. Tot desinteressadament, sense cap benefici econòmic.

Ara bé, com en els altres dos SO anteriors existeixen uns inconvenients:

1) Codi variable

Com que qualsevol usuari pot modificar el codi, es poden produir incompatibilitats a l'hora de canviar a una nova versió.

2) Solució problemes

Els problemes els ha de solucionar el propi usuari ja que no hi ha cap empresa al darrere que dongui suport a aquest SO.

3) Drivers i d'altres

Tots els programes i drivers que es volen instal·lar no tenen una versió per Linux. Poc a poc, aquest inconvenient va desapareixent ja que la majoria de programes venen amb Windows, Mac OS i Linux.

Per tant, a nivell d'usuari no acaba de ser un bon SO degut a que per fer-ne ús, es necessiten uns coneixements mínims de programació i del llenguatge (C, Python i d'altres), ara bé, conté una gran comunitat i és una eina completament gratuïta.

A nivell de SO dedicat a una aplicació concreta, com en el cas del Guanay, resulta el SO més adient a nivell de microprocessador.

Si es vol implementar en un microcontrolador s'haurà de tenir en compte la pròpia arquitectura d'aquest, les necessitats de l'usuari i la aplicació per la qual es vol fer servir.

### 2.2.2. SO per microcontroladors

#### Free RTOS



És un SO dedicat especialment per a microcontroladors [16], no obstant, no tots els microcontroladors poden suportar el SO, dependrà de les especificacions del mateix microcontrolador.

És un SO que es basa en la distribució de tasques per prioritats, tal com faria un SO ja creat com Windows10, mac OS o Linux. Això permet distribuir les tasques lliurement, com l'usuari desitgi.

Si bé és un avantatge per sistemes no gaire complexos, és costós quan es necessita un SO que realitzi més tasques amb cents de prioritats.

En la següent taula es mostren uns quants microcontroladors que admeten aquest sistema operatiu. A la pàgina oficial de freertos es poden trobar tots els models compatibles. En general, els fabricants de microcontroladors solen posar en el datasheet l'admeten.

Companyia	Microcontrolador
Atmel	SAMV7, SAM3, SAM4, AT91
Microhip	PIC32MX, PIC24EP, CEC17xx, MEC17xx
NXP	LPC1500, LPC1700, LPC1800, LPC4000
Texas Instrument	RM48, TMS570, MSP430X, MSP432
Renesas	RZ/A1, RX700, RX100, RL78

Taula 2.1. Microcontroladors que admeten el SO FreeRTOS.

## Linux

Com s'ha esmentat al apartat anterior. Linux i els seus derivats poden funcionar en microcontroladors que dependrà de les especificacions del fabricant d'aquest.

## 2.3. Anàlisi de Hardware

En aquest apartat es parlarà del hardware actual que podem trobar en qualsevol unitat de control.

### 2.3.1. FPGA vs Microcontrolador vs Microprocessador

La diferència principal que hi ha entre un microcontrolador, un microprocessador i una FPGA (Field Programmable Gate Array) és que les dues primeres són seqüencials i la darrera és combinacional (executa tot el codi alhora).

A continuació, s'explica les diferents característiques de cadascun d'ells.

### FPGA (Field Programmable Gate Array)

Una FPGA és un dispositiu lògic programable que conté blocs de lògica (hardware) on la seva interconnexió i funcionalitat es programa mitjançant un llenguatge de descripció especialitzat, normalment Verilog o VHDL, on el codi s'executa combinacionalment.

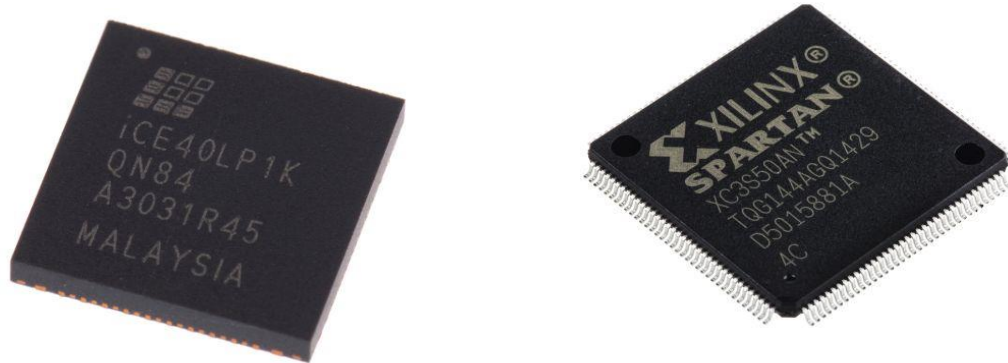


Figura 2.3. FPGA ice40LPKQN84 de Lattice (esquerra) i FPGA spartan XC3S250AN de Xilinx (dreta) [18].

La memòria d'aquests dispositius sol ser externa ja que està inclosa dins el mateix encapsulat, ara bé, hi ha fabricants que inclouen una memòria interna.

No té una estructura interna pròpia sinó que és l'usuari qui crea la seva pròpia estructura.

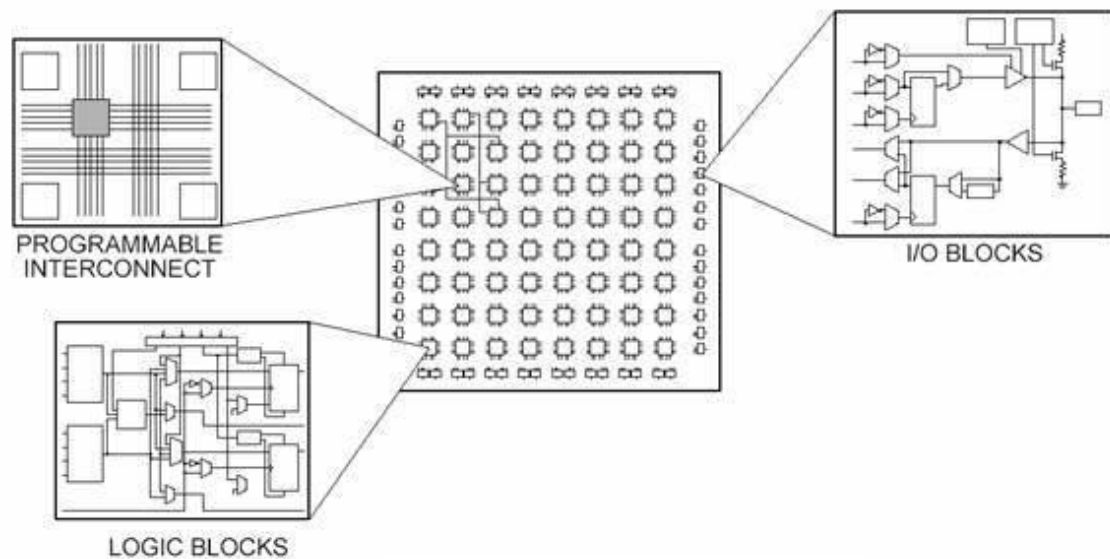


Figura 2.4. Estructura interna d'una FPGA [17].

Per tant, són ideals per fer rèpliques de microcontroladors que es volen testar abans de sortir al mercat.

Les eines per compilar el codi a les FPGA no són Open Source, per tant, no faciliten el lliure aprenentatge i obliguen a pagar llicències o a canviar de versions de programa, que després no són compatibles unes amb les altres.

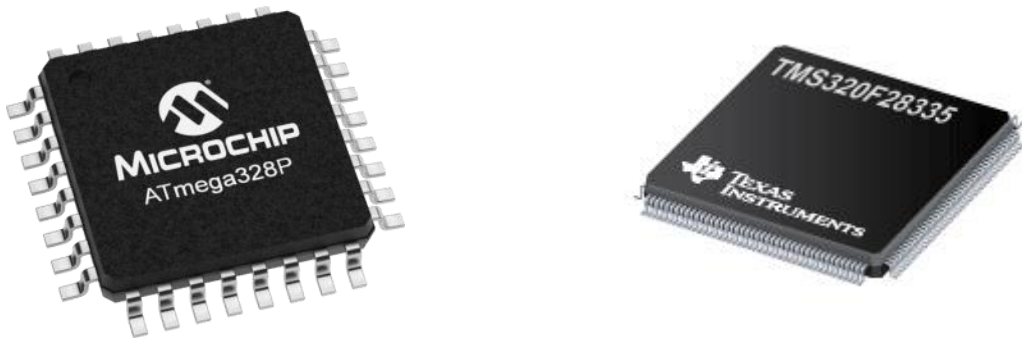
Al ser dispositius més complexos i no Open Source, tenen un seguiment menys extens per la comunitat d'usuaris. Per tant, compten amb pocs drivers, llibreries, foros, etc.

En termes econòmics, les FPGA són uns dispositius més cars [18] que els microcontroladors.

El seu ús sol estar més reservat a aplicacions concretes i crítiques.

### **Microcontrolador**

Un microcontrolador és un circuit integrat programable, capaç d'executar ordres dins la seva pròpia memòria o bé una memòria externa. El llenguatge de programació és variat i extens i les plataformes per programar depenen de cada fabricant. El codi s'executa seqüencialment.



*Figura 2.5. Microcontrolador ATmega328P [19] de Microchip (esquerra) i microcontrolador TMS320F28335 [20] de Texas Instrument (dreta).*

Els llenguatges de programació, en microcontroladors, més utilitzats són C/C++ i Python.

Algunes de les plataformes més utilitzades per programar els microcontroladors són:

MPLab (Microchip), Atmel Studio (Atmel), Energia (Texas Instruments) i Arduino (plataforma lliure Open Source).



Els microcontroladors tenen una arquitectura interna predefinida per cada fabricant.

En la Figura 2.6 es mostra l'arquitectura del microcontrolador ATmega328p del fabricant Atmel.

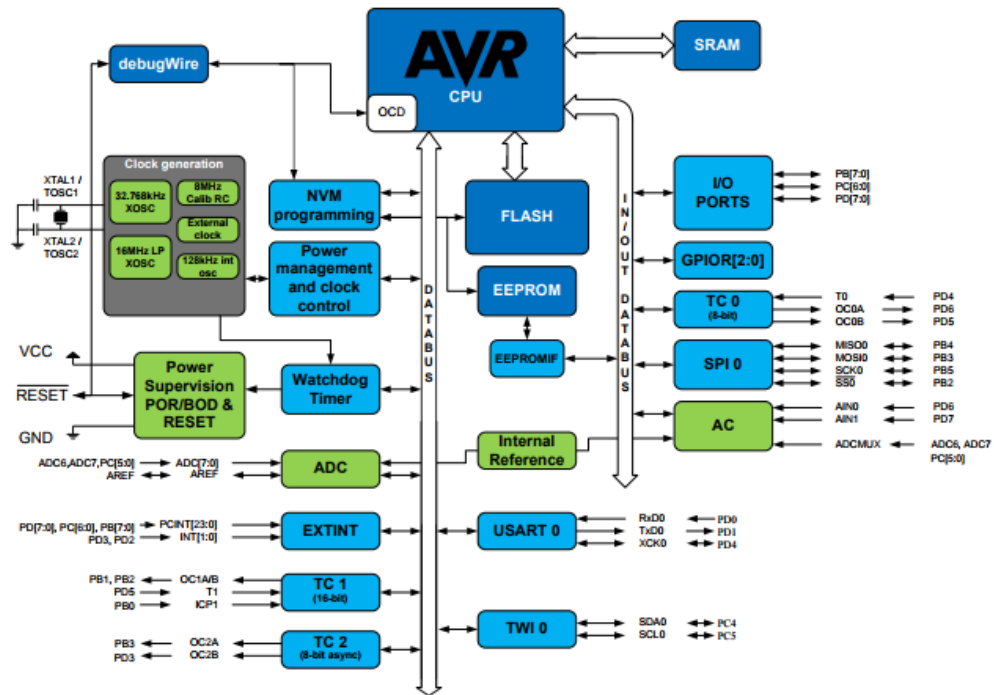


Figura 2.6. Arquitectura del microcontrolador ATmega328p [19].

Actualment, l'arquitectura més utilitzada és la ARM deguda al seu alt rendiment, el baix consum elèctric i el seu preu.

En la Figura 2.7 es mostra l'arquitectura del microcontrolador LM4F120 del fabricant Texas Instrument.

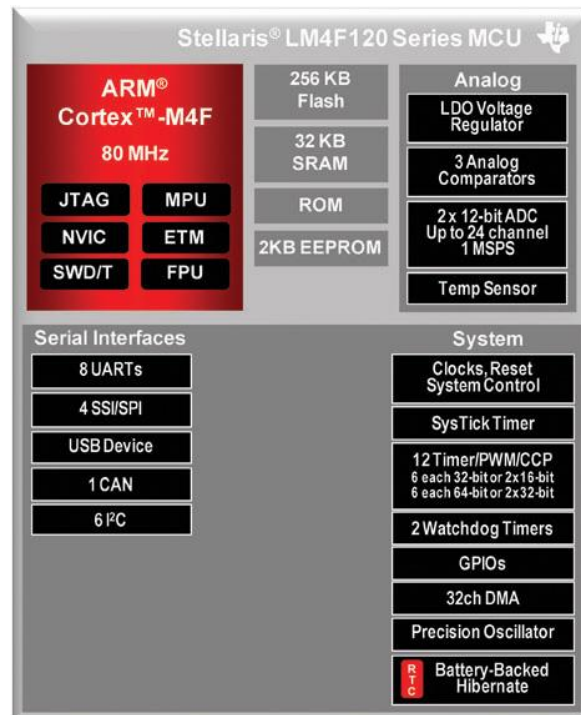


Figura 2.7. Arquitectura del microcontrolador LM4F120 [21].

En la següent figura, se situa la posició dels dispositius amb arquitectura ARM respecte els microcontradors i microprocessadors.

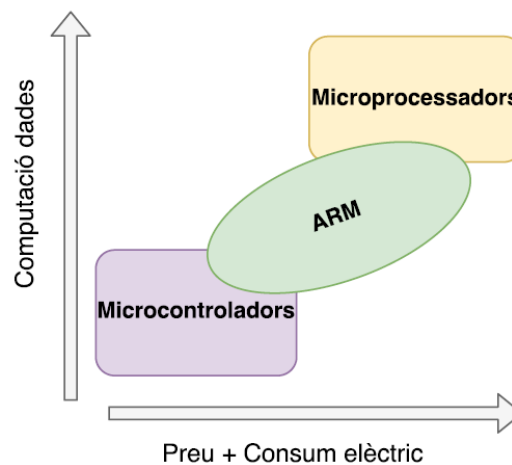


Figura 2.8. Posicions dels diferents sistemes segons la computació de dades, preu i consum elèctric.

L'ús dels microcontroladors està dedicat al control d'autòmates o dispositius en general que no necessitin un alt processament i tractament de dades. Per això últim, seria més convenient utilitzar els microprocessadors, que s'explicaran més endavant.

Els microcontroladors són uns dispositius que compten amb unes eines open source que han permès un increment del coneixement i d'usuaris en aquest sector. Hi ha infinitat de drivers, llibreries, exemples, informació i comunitats (foros) dedicat a aquest sector.

Un projecte que ha obert el sector dels microcontroladors a un públic més extens és Arduino. Arduino és una plataforma Open Source lliure, des de la interfície per programar fins el hardware de la placa, que porta el microcontrolador Atmel328p.

El ranking actual per ordre de fabricants de microcontroladors (MCU) són Renesas Electronics Corporation, NXP i Microchip (va comprar Atmel).

En la Figura 2.9 es mostra una gràfica de les empreses principals i la seva quota de mercat en el sector dels microcontroladors (MCU).

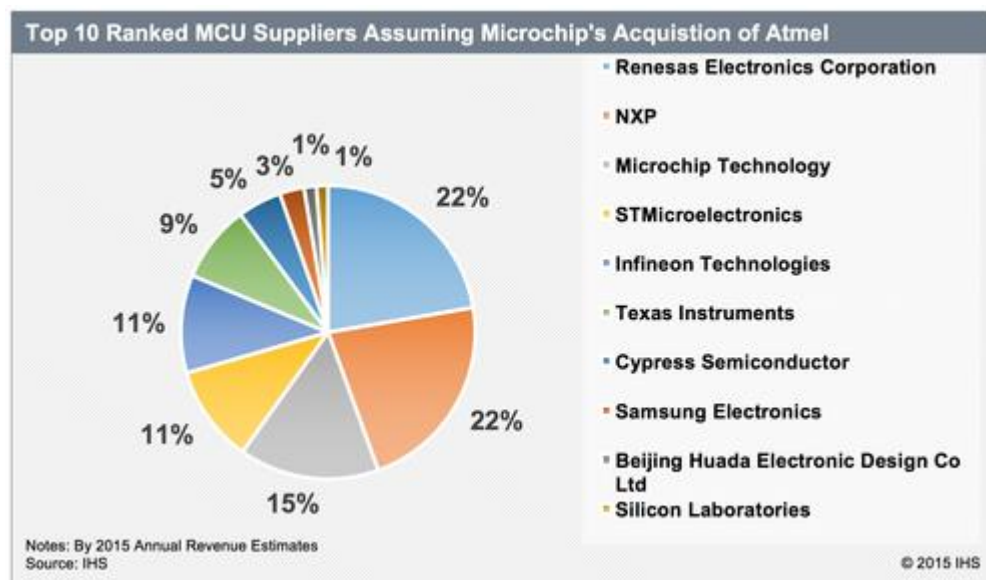


Figura 2.9. Gràfic de proveïdors de microcontroladors a nivell mundial el 2015 {22}.

### Microprocessador

Un microprocessador és un dispositiu encapsulat en un circuit integrat. És la unitat central més complexa d'un sistema informàtic.

No té memòries internes com un microcontrolador, sinó que han de ser externes i comunicades a través de busos. Això és degut a que el microprocessador està dissenyat per a gestionar una gran quantitat de dades, no per enmagatzemar-ne.

Cal tenir en compte que el microprocessador és un dispositiu seqüencial molt més ràpid que un microcontrolador. Actualment els microprocessadors "normals" d'ordinador van a 3 GHz, en canvi un microcontrolador sol anar entre uns 1kHz a 300 MHz. Està clar, que això depèn de cada model de microcontrolador i de microprocessador.

El microprocessador s'encarrega d'executar els programes i les aplicacions d'usuari, de gestionar dades i de prendre decisions.

Té una arquitectura formada per una o varies unitats central de processament (CPU).

La CPU es compon de:

Registres, Unitat de control, Unitat Aritmètica Lògica (ALU) i una unitat de càlcul.

En la Figura 2.10 es mostra l'arquitectura d'un ordinador amb una CPU, basada en el model de Von Neumann.

Aquests estan comunicats a través de busos a altres components com les memòries, unitats d'entrada i sortida (E/S) i d'altres.

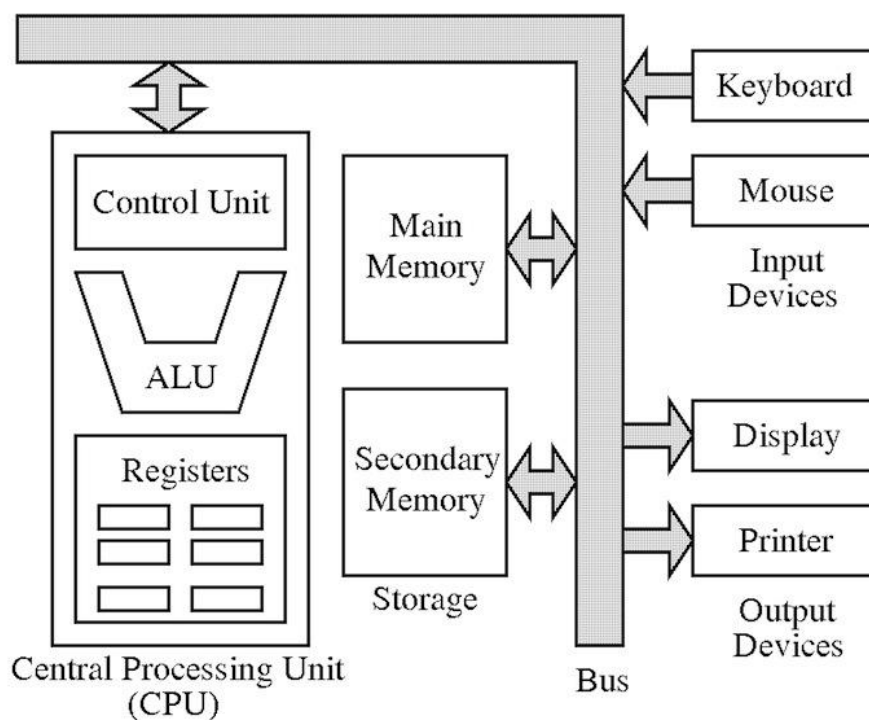


Figura 2.10. Arquitectura de Von Neumann [23].

Així doncs, la utilitat de les tres plataformes, FPGA, microcontroladors i microprocessadors, dependrà de les necessitats de l'usuari, tant a nivell de processament de dades, econòmic, recursos de software i d'altres paràmetres que s'han esmentat anteriorment.

### 2.3.2. Sistemes "Embedded" basats amb ARM

A continuació es parlarà de dos dispositius amb arquitectura ARM i que han estat objecte d'estudi.

#### **Raspberry Pi 3 Model B**

És una computadora (microprocessador) de placa reduïda [24] de baix cost desenvolupat al Regne Unit per la Raspberry Pi Foundation, amb l'objectiu de facilitar l'aprenentatge a les escoles i universitats, en el camp de l'electrònica i computació.

Aquest dispositiu compta amb un software open source, en concret compta amb una versió adaptada de Debian (GNU/Linux) [25], però també compta amb sistemes operatius més concrets especialitzats amb Smart TV, Videojocs o d'altres. Depèn de la utilitat que li vulgui donar l'usuari s'utilitzarà un SO o altre.

Cal tenir en compte que haurà de ser un SO lleuger, és a dir, que ni Windows ni molt menys Mac OS podrà funcionar.

El hardware en canvi, és privatiu.

En la següent taula es pot veure les principals diferències entre la Raspberry Pi 2 Model B (Figura 2.11. a l'esquerra) i la Raspberry Pi 3B (Figura 2.11 a la dreta).



Figura 2.11. Raspberry Pi 2 Model B (esquerra) i Raspberry Pi 3 Model B (dreta).

	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B
<b>Any</b>	2014	2016
<b>SoC</b>	Broadcom BCM2835	Broadcom BCM2837
<b>CPU</b>	700 MHz ARM1176JZF-S	1,2GHz QUAD ARM Cortex-A53
<b>RAM</b>	512 Mb	1 Gb
<b>ROM</b>	SD	MicroSD
<b>Pins</b>	26	40
<b>USB</b>	2 x 2.0	4 x 2.0
<b>Video</b>	RCA, HDMI	Jack, HDMI
<b>Xarxa</b>	Ethernet 10/100	Ethernet 10/100, WiFi, Bluetooth
<b>Consum</b>	700 mA / 3,5W / 5V	2,5 A / 12,5 W / 5 V
<b>Tamany</b>	85,6 x 53,98 mm	85 x 56 mm
<b>Preu</b>	35 €	35 €

Taula 2.2. Característiques de Raspberry Pi 2 B i Raspberry Pi 3 B.

En concret, la disposició dels pins de la Raspberry Pi 3 Model B és la següent:

Raspberry Pi 3 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 2  
29/02/2016

www.element14.com/RaspberryPi

Figura 2.12. Disposició dels pins de la Raspberry Pi 3 B [26]..

Les entrades GPIO es podran configurar com a ports d'entrada i/o sortida.

També mitjançant aquests pins i diverses llibreries es poden establir diferents funcions com el PWM (Pulse Width Modulation) o protocols com I2C i SPI.

La Raspberry Pi és un dispositiu àmpliament testejat, que compta amb una gran comunitat i multituds de webs, foros i altres on es pot trobar informació, projectes, exemples, drivers, llibreries i més.

### BeagleBone Black

És una computadora (microprocessador) de placa reduïda [27] creada per la comunitat Beagleboard.org totalment lliure (Open Source) tant en hardware com en software derivada del anterior model BeagleBone.

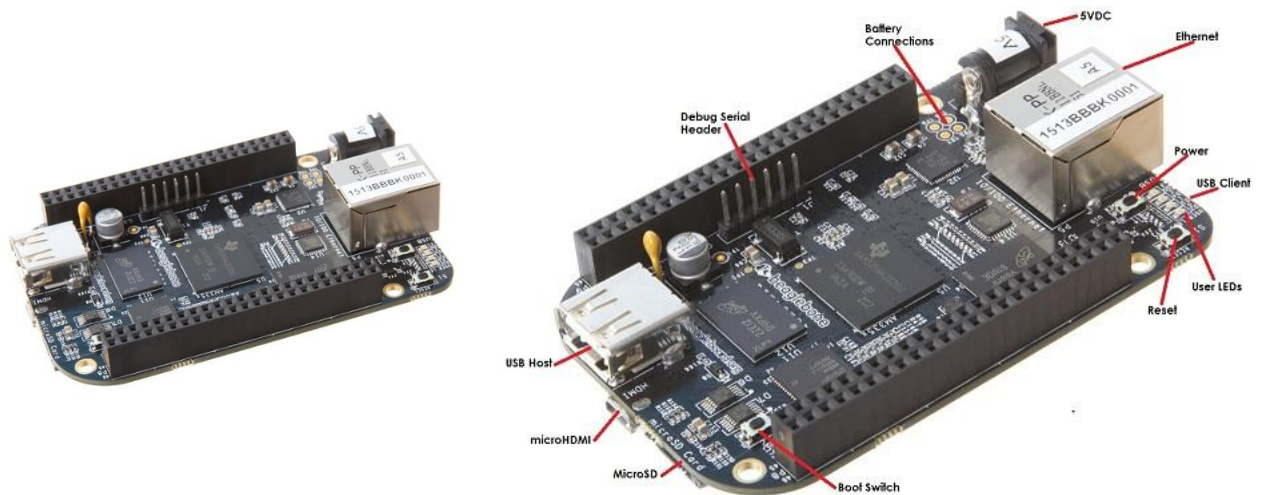


Figura 2.13. BeagleBone Black.

Aquest dispositiu compta amb un SO precarregat. Tot i que posteriorment, es pot treballar amb altres SO com Android, Debian i Ubuntu entre d'altres.

En la següent taula es mostren les característiques principals d'aquesta placa.

	BeagleBone Black
Any	2014
Processador	Sitara AM3358BZCZ100 1 GHz, 2000 MIPS
Onboard Flash	4 Gb, 8 bit
SDRAM Memory	512 Mb, 800MHz



ROM	miniSD
Pins	92
USB	1 x 2.0
Video	HDMI
Xarxa	Ethernet 10/100
Consum	210-460 mA / 2,3 W / 5V
Tamany	85,6 x 53,98 mm
Preu	55 €

Taula 2.3. Característiques de la BeagleBone Black.

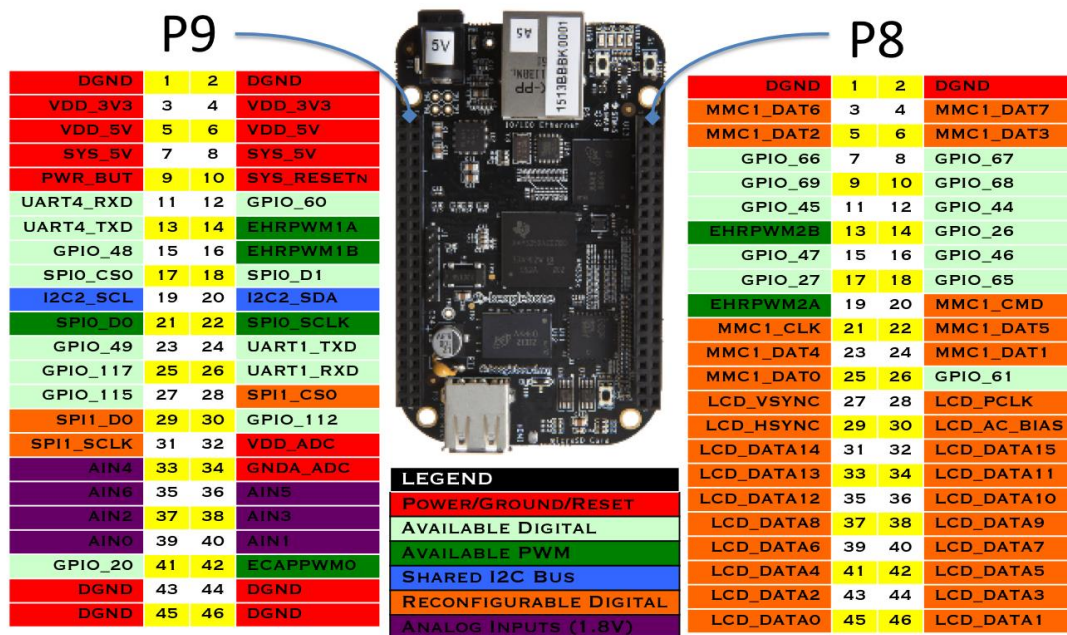


Figura 2.14. Pins de la BeagleBone Black [27].

En la següent figura es mostren els pins de la placa.

Les entrades GPIO es podran configurar com a ports d'entrada i/o sortida.

També mitjançant aquests pins i diverses llibreries es poden establir diferents funcions com el PWM (Pulse Width Modulation) o protocols com I2C i SPI.

A diferència de la Raspberry Pi (qualsevol model) té molta més varietat en quant a pins. Ens permet tenir protocols com I2C, SPI i CAN. Aquest últim protocol és suportat per poques plaques de desenvolupament (Raspberry Pi no l'admet) i és un estàndard molt utilitzat en automoció ja que és un protocol altament robust.

Com la Raspberry Pi, és un dispositiu àmpliament testejat, que compta amb una gran comunitat i multituds de webs, foros i altres on es pot trobar informació, projectes, exemples, drivers, llibreries i més.

És una placa ideal per computar i gestionar gran quantitat de sensors degut als seus pins i la seva alta capacitat de computació.

### **2.3.3. Miniordinadors**

Els miniordinadors o miniPC són ordinadors que tenen les mateixes (o més) prestacions que els PC de sobretaula i portàtils.

El principal avantatge d'aquest miniordinador, com bé diu la paraula, és que són minis, i per tant, poden estar a llocs amb un espai reduït. El convertidor AC/DC principal és extern i per tant no ocupa ni espai ni consum dins del pròpi PC.

Actualment, aquests ordinadors estan dissenyats per albergar discs durs en estat sòlid, és a dir, discs durs que ja no compten amb un disc dur físic, i per tant, tots els inconvenients que podien sorgir. Com per exemple que no es poden moure massa i això per un aparell autònom com pot ser el Guanay, és un inconvenient més que notable.

En la següent taula es mostra una comparació entre els dos miniPC.



*Figura 2.15. Intel NUC [28] (esquerra) i Asus VivoMini [29] (dreta).*

	Intel NUC 5i7RYH	Asus VivoMini UN65
<b>Any</b>	2015	2015
<b>ROM</b>	HDD 50-3000 GB	SSD 32-256GB
<b>RAM</b>	Màx. 32 GB	4-16 GB
<b>Processador</b>	Intel Core i7	Intel Core i7-6500U
<b>Sistema operatiu</b>	Windows 10/Linux	Windows 10/Linux
<b>Freqüència</b>	3,4 GHz	3,1 GHz
<b>Pins per controlar</b>	NO	NO
<b>Xarxa</b>	WiFi / Ethernet	WiFi / Ethernet
<b>USB</b>	4 x USB 3.0	2 x USB 3.1 2 x USB 3.0
<b>Consum</b>	12-19 Vdc / 28 W	2,5 A / 125 V
<b>Tamany</b>	116,6 x 112 mm	131 x 131 mm
<b>Preu</b>	550 € [28]	500 € [30]

Taula 2.4. Característiques de Intel NUC5i7RYH i de Asus VivoMini UN65.

Ja que són sistemes pensats per moure una gran quantitat de dades, s'hi poden incloure programes que requereixen més ROM, RAM i tarjeta gràfica. Com per exemple programes d'anàlisis d'intel·ligència artificial i programes de càlcul. No obstant això, apareixen un seguit de problemes.

Com es veu a la figura (sobre gràfic ARM vs microc vs microp), els microcontroladors que estan inclosos en un miniPC, tenen un consum i preu molt superior als microcontroladors "standard" i als microcontroladors/processadors que disposen d'una arquitectura ARM.

Per tant, s'ha d'anar en compte amb l'ús d'aquests dispositius. En ocasions aporten més problemes que avantatges sobretot en l'autonomia de bateries i els pins nuls pel control i/o sensors.

### 2.3.4. Microcontroladors Atmega328 vs MC9S08PT32

El camp dels microcontroladors és molt extens. Per tant, en aquest apartat se centrarà en dos models de dues marques diferents de microcontroladors de 8 bits.

En la següent taula es pot veure els milions de dòlars que mou cada tipus de microcontrolador.

	2012	2013	2014	2015	2016	2017	CAGR
Total Semiconductor	325,367	339,666	361,612	385,052	395,974	413,602	4.9%
Microcontroller (MCU)	16,008	16,202	17,211	18,799	19,307	20,480	5.1%
4-bit MCU	154	159	161	157	145	133	-2.8%
8-bit MCU	6,057	6,565	6,936	7,532	7,768	8,259	6.4%
16-bit MCU	4,021	3,611	3,765	4,060	4,053	4,019	0.0%
32-bit MCU	5,776	5,868	6,349	7,050	7,341	8,069	6.9%

Taula 2.5. Venta de microcontroladors (en milions de dolars) [19].

Actualment, els microcontroladors de 8 bits encara segueixen sent molt usats ja que en moltes aplicacions no són necessàries més prestacions.

Altres avantatges que ens aporten els microcoontroladors de 8 bits són: el seu baix consum energètic, cost i la seva gran popularitat i robustesa.

#### ATmega328P

És un microcontrolador de 8 bits de la marca Atmel (recentment comprada per Microchip [31]) que compta amb plataformes de programació com Atmel Studio i Arduino (Open Source).

També té una extensa comunitat, documentació, llibreries, drivers, projectes i més.

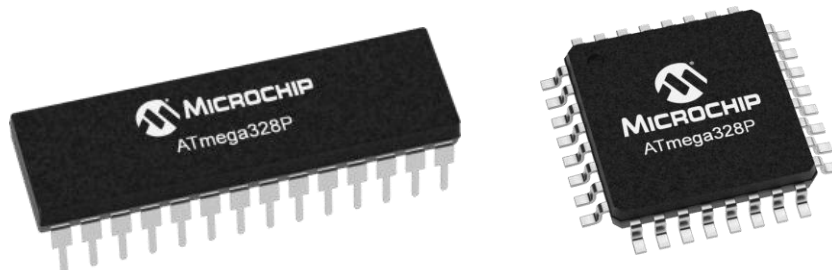


Figura 2.16. Microcontrolador ATmega328P [19].

### MC9S08PT32

És un microcontrolador de 8 bits de la marca NXP [32]. Té un ús força extens en automoció i electrònica domèstica.

Compta amb plataformes com FreeMaster i LPCXPRESSO [33].



Figura 2.17. Microcontrolador MC9S08PT32 [34].

En la següent taula es realitza una comparació dels dos microcontroladors.

	ATmega328P	MC9S08PT32
Program Memory Type	Flash	Flash
Program Memory (KB)	32	32
CPU Speed	20 MIPS – 1MHz to 20 MHz	8 MHz to 40 MHz
RAM Bytes	2048	4096
Data EEPROM (bytes)	1024	256
Digital Communication Peripeherals	1-UART,2-SPI,1-I2C	3-UART; 2-SPI
ADC	8 bit out – 10 bit in	12 bit
Capture/Compare/PWM Peripherals	1-Input Captures, 1-CCP, 6- PWM	16-PWM
Timers	2x 8-bit, 1 x 16-bit	1 x 16-bit
Comparators	1	1

Temperature rang ( Cº )	-40 to +105	-40 to +105
Operating Voltage Range (V)	1,8 to 5,5	2,7 to 5,5
Pin Count	32	32
Low Power	Yes	Yes
Price	2,5 €	1,8 €

Taula 2.6. Característiques del ATmega328P i del MC9S08PT32.

En aquest projecte, com s'explicarà en el CAPITOL 5. S'ha triat el ATmega328P per la facilitat que té per disposar de recanvis, llibreries, drivers i documentació.

## 2.4. Futur del mercat

Actualment el mercat no para d'expandir-se, reduint tamanys i preus, i augmentant capacitats de processament, memòries RAM i ROM.

L'estudi que s'ha fet en aquest apartat en uns 20 anys serà completament diferent, s'ha de tenir en compte que en aquest apartat del treball s'han analitzat les tecnologies actuals tant en sistemes operatius com en plaques de hardware.

Ara bé, cap on sembla que tendeix el mercat?

Els sectors que estàn patint un augment d'electrònica són el sector industrial, telecomunicacions i sobretot l'automòbil, aquest últim gràcies als automòbils elèctrics.

Un factor important que s'ha de tenir en compte és la obsolescència programada de molts components electrònics, fabricats perquè en uns anys deixin de funcionar. Això permetrà reemplaçar aquests components o fabricar-ne de nous, però també generant una gran quantitat de residus electrònics.

Si es parla de microcontroladors, no es pot deixar d'anomenar dues tecnologies que està entrant en el mercat, els SoC i el IoT.

Els "System on Chip" (SoC) [35] són bàsicament microcontroladors dedicats a una aplicació específica. Per exemple, hi ha models que tenen integrats un mòdul WiFi dins el pròpi encapsulat.

En la següent figura es veu un SoC de la marca Nordic Semiconductor. Aquest SoC inclou un mòdul bluetooth i té un baix consum.

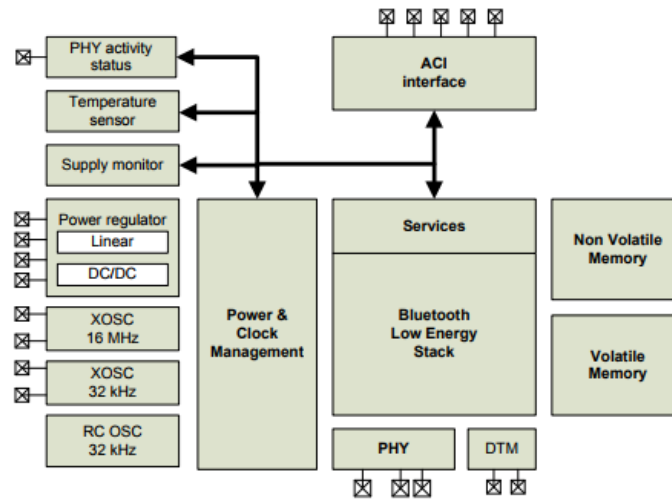


Figura 2.18. Diagrama de blocs del SoC de la marca Nordic Semiconductor [36].

L' "Internet of Things" (IoT) [37] és un concepte de connectar diferents dispositius a internet. Per exemple, per disposar d'una casa domòtica.

També sembla una aplicació interessant en el cas dels AUV, ja que es pot controlar el AUV desde el mòbil mitjançant una xarxa d'internet.

A continuació i per finalitzar es mostra una gràfica del creixement de les aplicacions del IoT i la seva futura expansió.

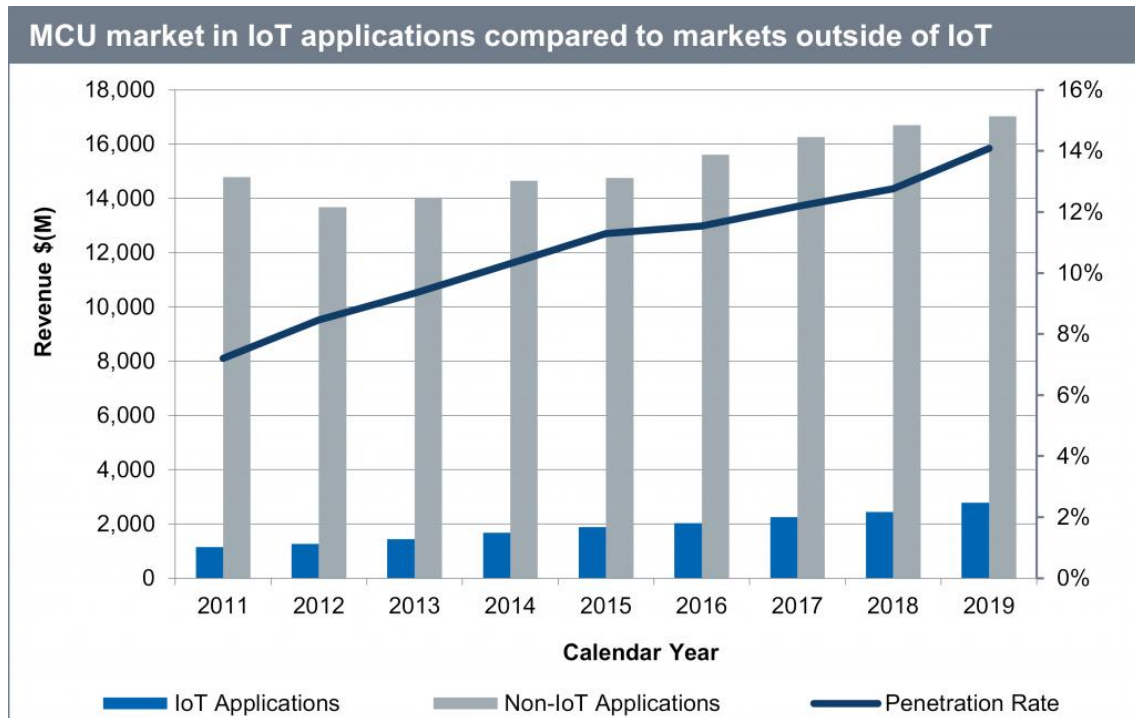


Figura 2.19. Gràfic del mercat del IoT [38].



## **CAPÍTOL 3. GUANAY 2.1**

L'objectiu d'aquest capítol és fer una adaptació d'una nova unitat de control que substitueixi l'actual PC104+ incorporat al Guanay II.

Aquest capítol sorgeix de la necessitat d'incorporar una nova unitat de control degut a la falta de prestacions del PC104+ i que a més possibiliti noves aplicacions i incorporacions de nous instruments en el vehicle com un sonar (Sound Navigation And Ranging) o un Doppler Velocity Logs (DVL).

L'objectiu principal és que es pugui substituir el PC104+ en cas de emergència sense ser intrusius amb el sistema actual, podent aprofitar la plataforma Labview utilitzada fins ara en el Guanay II.

### **3.1. PC104+**

És la unitat de control principal del Guanay II.

El PC104+ [39] és un PC "Embedded" [40] que integra diferents mòduls per aplicacions, RS232 i pins per interactuar amb actuadors i sensors.

És una placa de fa més de 20 anys, així doncs, s'entenen les limitacions que pot tenir avui en dia degut a l'avanç tecnològic.



*Figura 3.0. PC104+ [39] (esquerra) i Intel NUC5i7RYH [28] (dreta).*

A continuació, a la Taula 3.0 es mostra una comparativa [28][41][42] entre les dues unitats de control principals, la del Guanay II basada en el PC104+ i la del Guanay 2.1 basada en la Intel NUC5i7RYH.

	<b>PC 104+</b>	<b>Intel NUC5i7RYH</b>
Any	1992	2015
ROM (GB)	Flash 1-8	HDD 50-3000
RAM	512 MB	Max. 16 GB
Processador	AMD Geode LX800	Intel Core i7
Sistema operatiu	Windows XP	Windows 10/Linux
Freqüència	500 MHz	3,4 GHz
Pins per controlar	SI	NO
Voltatge (VDC)	+12/5	19/12
Potència (W)	6	28
Wifi	NO	SÍ
Ethernet	SÍ	SÍ
USB	2 x 2.0	4 x 3.0
Pes (kg)	0,13	0,690
Espai (mm)	95,89 x 90,17	116,6 x 112
Preu (€)	350	550

Taula 3.0. Comparativa entre PC104+ i Intel NUC5i7RYH.

Les prestacions d'ambdues plaques són completament diferents on cadascuna té les seves avantatges i inconvenients en funció de la aplicació. Per exemple:

- **Processador:**  
En aquest cas, el processador de la NUC5i7 (Core i7) que treballa a 3,4 Ghz és molt millor que el del PC104+ (Geocode LX800) a 500 MHz.
- **Pins:** El PC104+ té pins per col·locar diferents dispositius, ports de comunicacions, extensions (shields) entre d'altres, mentre que la NUC5i7RYH només disposa de ports USB.
- **Potència:** La NUC5i7RYH té un consum d'uns 28 W com a màxim, mentre que el PC104+ té un consum d'aproximadament uns 6 W.
- **Pes:** La NUC5i7RYH pesa uns 700 grams, mentre que el PC104+ pesa uns 130 grams.
- **Preu:** En aquest cas, els preus d'ambdós és força variable ja que depèn de les característiques finals del dispositiu, plaques d'expansió (PC104+), lloc de venda entre d'altres. Tot i així, els preus aproximats serien uns 350 € per el PC104+ i 550 € per la NUC5i7.

Les característiques de la NUC5i7RYH són admissibles pel tipus d'arquitectura (bateries) i estructura mecànica (pes) actual del Guanay II.

### 3.2. Unitat de control del Guanay II i Guanay 2.1

A continuació s'analitza l'arquitectura del Guanay II i del Guanay 2.1.

#### 3.2.1. Estructura de control del Guanay II

Aquesta arquitectura va ser redissenada per Ivan Masmitjà i J.Gonzalez [1][2], on es va incloure un microcontrolador PIC (per l'accionament dels motors de direcció) entre d'altres elements.

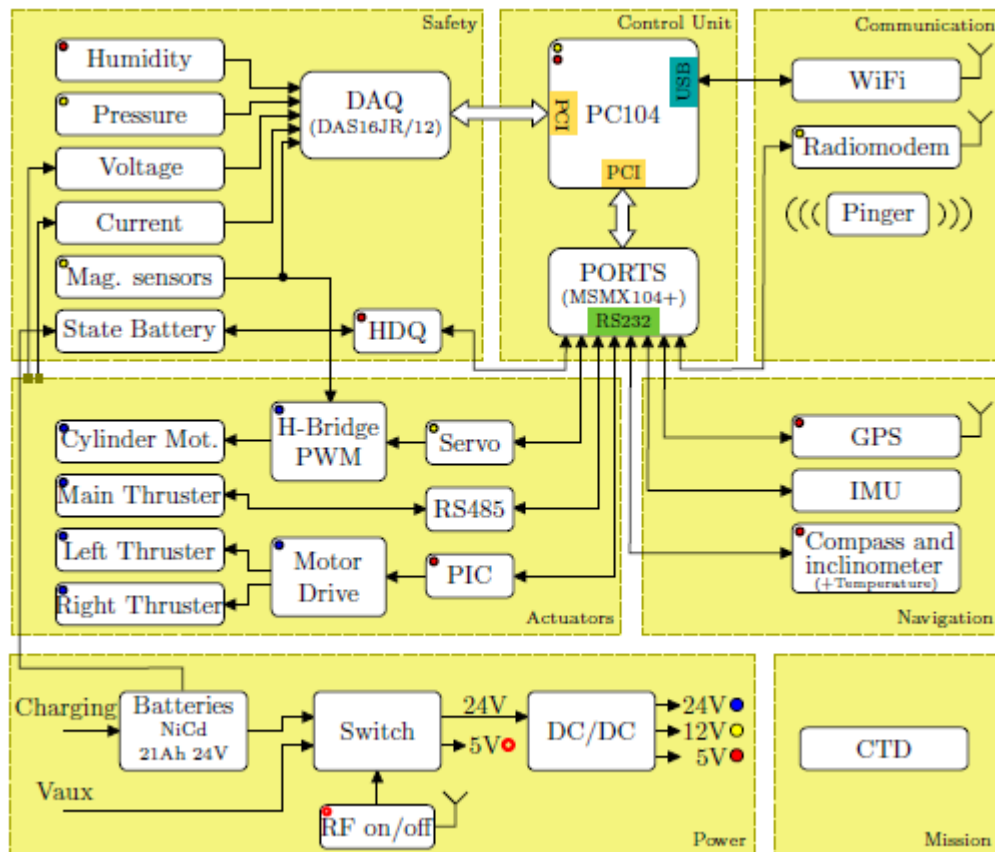


Figura 3.1. Arquitectura Guanay II.

Com es pot veure en la Figura 3.1, el Hardware del Guanay II està distribuït en sis blocs:

**Control Unit:** Unitat de control del Guanay II. És el sistema que permet obtenir i recopilar totes les dades, tractar-les i gestionar-les.

En aquest apartat es consideren també els diferents “hubs” que hi ha per transmetre les dades dels actuadors o sensors al PC104+. En aquest cas, el “hub” és el PORT MSXM 104+ [1] que té uns connectors RS232.

**Safety:** Sistema de seguretat que permet adquirir dades mitjançant sensors i que a través d'un DAQ (convertors analògic a digital) model DAS16JR/12 [1] permet transmetre-les a la unitat de control (PC104+).

**Communication:** En aquest bloc s'hi troben les comunicacions que ens permeten estar connectats amb el Guanay II, tant enviant dades com rebre-les.

**Actuators:** Sistema que permet el control dels motors i del servomotor que mou el pistó que permet submergir o emergir el submarí.

En aquest apartat hi trobem el microcontrolador PIC16F1508 que permet generar senyals PWM i a través d'un pont en H permet moure els motors dret (Right Thruster) i esquerra (Left Thruster).

**Navigation:** Bloc on s'hi troben els diferents sistemes per saber la posició del submarí. També es pot trobar la velocitat i altres paràmetres que faciliten la geocalització.

**Power:** Sistema d'energia del Guanay II. S'hi troben unes bateries de NiCd a 21Ah i 24V amb una serie de convertors (DC/DC) que adapten el voltatge pels diferents dispositius del sistema.

També s'hi troba una antena RF HIRK-315AP [43] que mitjançant la radiofreqüència permet activar un interruptor ("switch") per engegar l'etapa de potència del Guanay II.

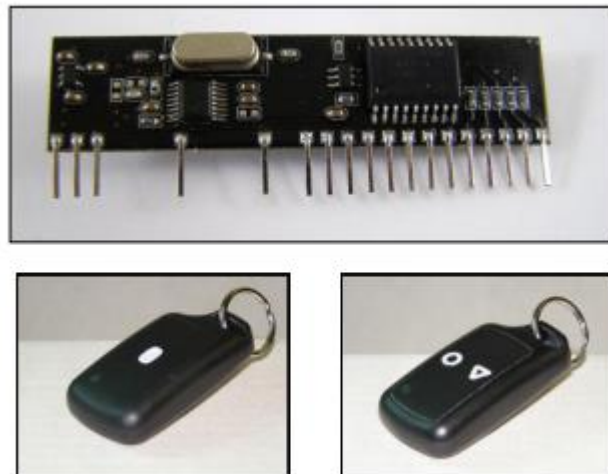


Figura 3.2. HIRK-315AP.

**Mission:** Hi trobem el CTD model XR-420 [44]. El CTD és un instrument que ens permet mesurar la conductivitat, temperatura i profunditat de l'aigua.



Figura 3.3. CTD XR-420 [44].

### 3.2.2. Estructura de control del Guanay 2.1

En aquest apartat, la unitat de control PC104+ és reemplaçada per una Intel NUC5i7RYH, així com els seus ports d'expansió ("hubs").

A continuació es mostra la nova estructura amb la unitat de control reemplaçada.

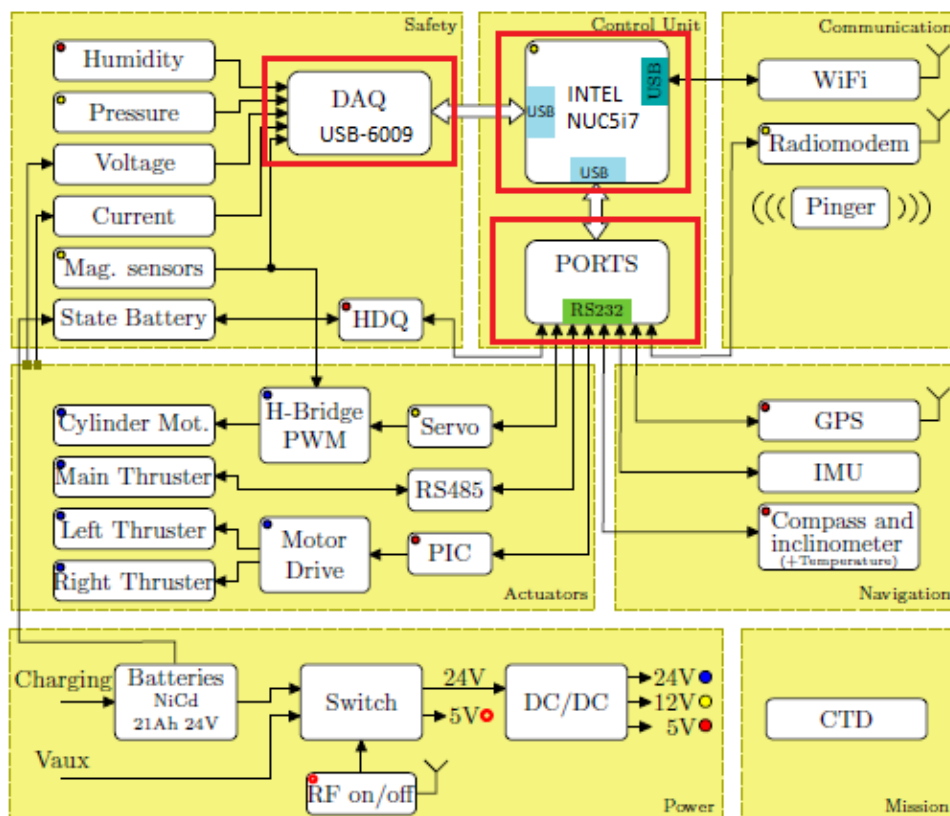


Figura 3.4. Arquitectura Guanay 2.1.

Com es pot veure amb vermell, s'ha reemplaçat la unitat de control principal (Control Unit), el sistema d'adquisició de dades (DAQ) per un USB-6009 i uns PORTS que adapten les entrades i sortides del port RS232 a un USB 2.0.

La diferència bàsica entre el DAQ USB-6008 i el USB-6009 [45] és que aquest últim té més resolució però també és més car.

En la pràctica que s'ha dut a terme en aquesta part del projecte, s'ha utilitzat el USB-6008 ja que és el que disposava el laboratori d'electrònica de la universitat.

També com es pot comprovar, la Intel NUC està alimentada a 12V mitjançant el DC/DC original (Mornsun series VRB) del Guanay II. Tot i que amb els valors teòrics funciona, s'hauria de provar al laboratori amb la NUC i el convertidor DC/DC.

### **3.3. Proves experimentals**

Per demostrar que pot executar-se el programa sense problemes primer de tot s'ha instal·lat el sistema operatiu Windows 10 i posteriorment els diferents programes.

Concretament, el Labview, que és el programa, basat amb objectes, amb el que està fet tot el sistema d'adquisició i tractament de dades perquè el Guanay II funcioni.

Com que no es disposaven de recanvis dels components que hi ha en el Guanay II s'han utilitzat dos sensors analògics.

Un sensor que mesura la temperatura (LM35) i un altre que mesura la pressió (RS PRO Industrial Pressure Sensor 7975046).

#### **3.3.1. Sensor de temperatura LM35**

És un sensor linial de temperatura analògic. El fabricant del model utilitzat es Texas Instruments.

Ens dona un rang de temperatura entre +2 i +100 °C amb una sortida de 0 a 5V si el sensor és alimentat a 5 Vdc.

El fabricant recomana una alimentació entre 4-20 Vdc.

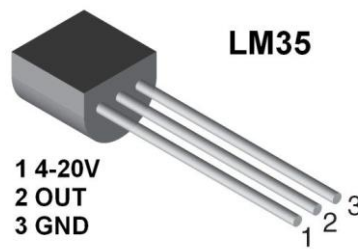


Figura 3.5. Sensor LM35.

Té una precisió de  $\pm 10 \text{ mV/}^\circ\text{C}$  amb una temperatura de funcionament entre  $-55 - +150^\circ\text{C}$ .

Tot i que el sensor sigui molt precís, posteriorment s'haurà de calibrar.

### 3.3.2. Industrial Pressure Sensor

És un transductor linial de pressió analògic [48] amb un microcontrolador incorporat. Té el codi de referència 797-5046 i és de la marca RS Pro [18].

Aquest model en concret mesura un rang entre 0 i 6 Bar G amb una sortida de 0 a 5 Vdc.

El voltatge d'alimentació d'entrada té un rang de 9 a 32 Vdc. Si es posa a menys voltatge no funcionarà correctament, com va passar quan es feien les proves en el laboratori.



Figura 3.6. Sensor de pressió.

Té una precisió de  $\pm 0,25\%$ , amb una temperatura de funcionament entre  $-20$  i  $+135^\circ\text{C}$ , i compta amb una carcassa d'acer inoxidable que ens permet posar-lo tant a l'interior com a l'exterior del submarí.

Compta amb quatre cables: Alimentació (9-32 Vdc), Massa (GND), Signal + i Signal - .



Per fer les mesures bàsiques només agafarem tres dels cables, en concret, l'alimentació, massa i signal +.

Tot i que sigui molt precís, posteriorment s'haurà de calibrar.

### **3.3.3. Introducció de la implementació amb Labview**

La instal·lació del sistema operatiu i diferents plataformes no han donat cap tipus de problema. S'ha instal·lat el Labview amb el "codi" que permet el funcionament del Guanay II.

Veient els requisits que demana Labview per ser instal·lat i la capacitat que té la NUC5i7RYH no hi haurà cap problema en fer anar el codi sense que es quedi bloquejat com passa ara en el PC104+.

Ja que pretén ser substituït el PC104+ per la NUC5i7RYH en cas d'emergència, el més important que s'hauria de tenir en compte és desactivar les actualitzacions del sistema operatiu Windows 10 (instal·lat actualment), i que un cop s'engegui tot el sistema operatiu, s'executi el Labview automàticament. Això últim, s'ha implementat amb la demostració que s'ha fet amb el sensor de temperatura i pressió que es veu en aquesta part del treball.

Per fer l'adquisició de dades s'ha utilitzat un DAQ de la marca Texas Instruments, en concret el model USB-6008 [45] que es mostra en la Figura 3.7.



*Figura 3.7. DAQ USB-6008 [45].*

Aquest dispositiu compta amb entrades i sortides digitals i analògiques, aquestes amb una resolució de 12 Bits.

Per identificar el USB-6008 al Labview es fa servir un objecte que s'anomena DAQ Assistant.

S'utilitzaran únicament dues entrades analògiques, en concret les AI 2 i AI 3, que es configuraran de manera que el DAQ adquireixi voltatge.

Per fer-ho s'ha seguit un manual de Texas Instruments i vídeos d'internet.

Labview [46] compta amb dues pantalles, una gràfica anomenada Frontal Panel (panell frontal) i l'altre que anomenada Block diagram (diagrama de blocs) on estàn els diferents objectes relacionats a través de línies.

A continuació s'inclou la Figura 3.8 on es visualitzen diversos gràfics que mostren les diferents dades que mesura el DAQ a través dels sensors.

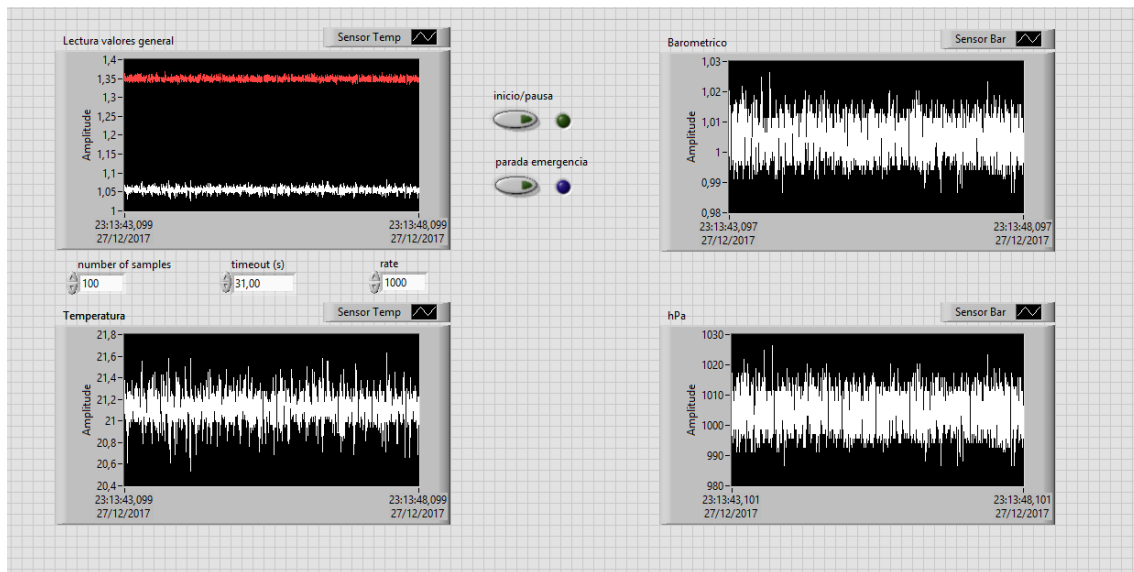


Figura 3.8. Panell frontal Labview. Programa parat.

En la Figura 3.9, es mostra el diagrama de blocs, que consta de un bucle while (rectangle exterior) i una estructura de casos (rectangle interior amb "True").

El bucle while ens permetrà que el programa funcioni constantment, excepte quan es premi un interruptor de parada d'emergència que farà parar el bucle (Figura 3.9).

L'estructura de casos ens permetrà, mitjançant un interruptor, deixar de mostrar les dades del DAQ en els gràfics.

Per últim, la Figura 3.10 mostra el programa en funcionament.

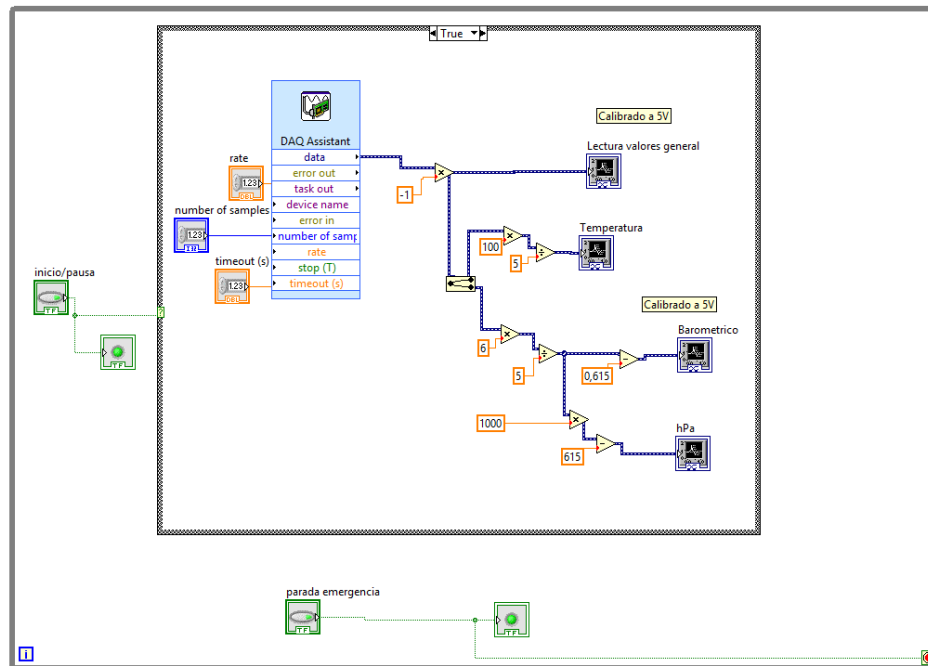


Figura 3.9. Diagrama blocs Labview. Programa parat.

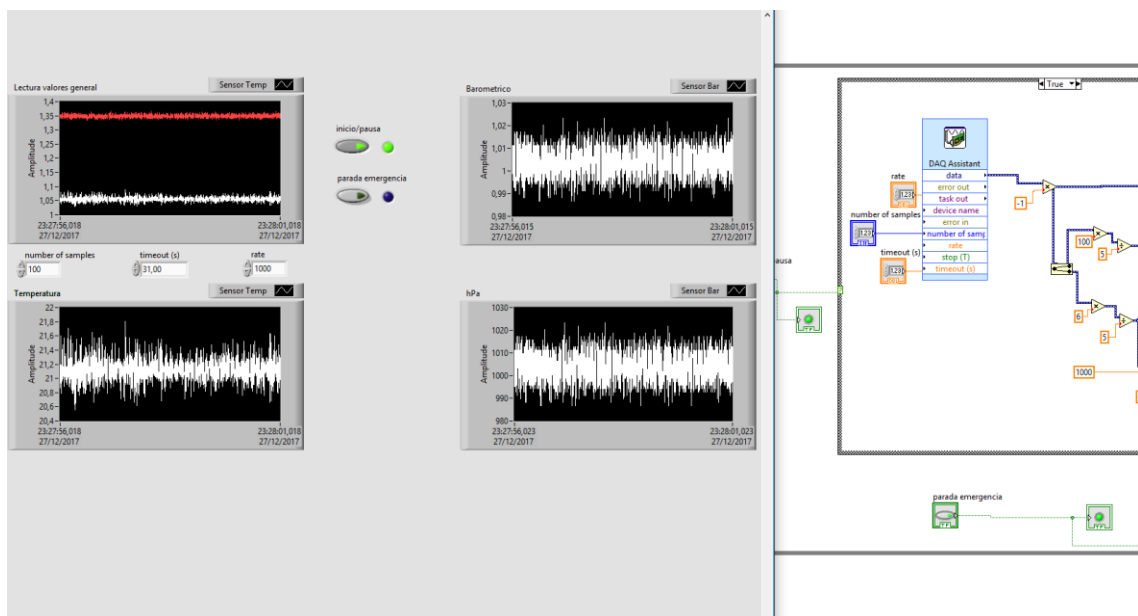


Figura 3.10. Panell Frontal (esquerra) i diagrama de blocs (dreta). Programa en funcionament.

### 3.3.4. Anàlisi detallat del programa

A continuació s'analitzen els diferents apartats del programa:

#### **Adquisició de dades**

S'ha fet servir un "cluster" anomenat DAQ Assistant on es poden ajustar els paràmetres per llegir a través del USB-6008.

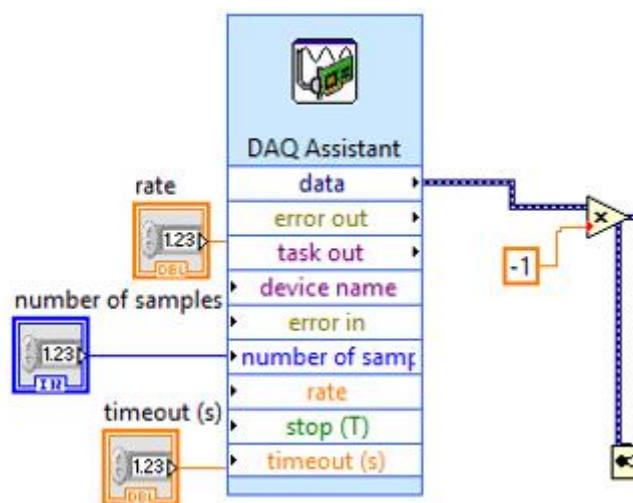


Figura 3.11. Diagrama blocs DAQ Assistant.

S'hi ha afegit un multiplicador -1 ja que els valors donaven inversos. Així, tots els valors queden positius.

### Gràfic de voltatge de temperatura i voltatge de pressió

S'afegeix un element per la mostra de dades a través d'un gràfic.

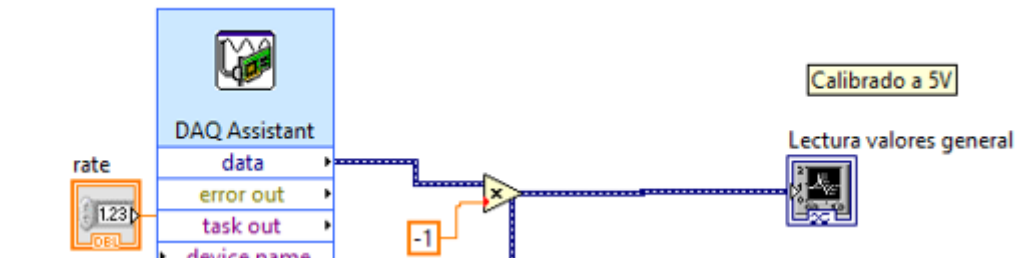


Figura 3.12. Diagrama de blocs amb gràfic.

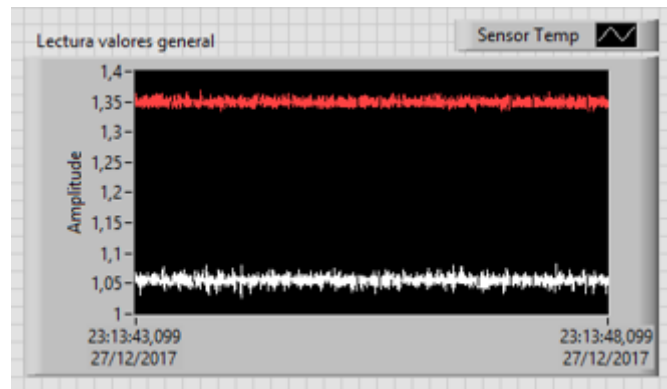


Figura 3.13. Lectura valors temperatura i pressio en volts.

### Gràfic Temperatura en graus (°C)

Per calibrar aquest sensor s'ha fet una regla de tres, tenint en compte les especificacions del sensor LM35.

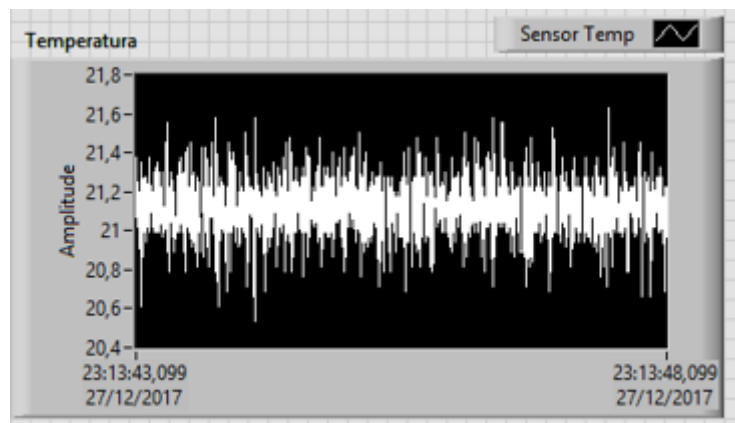


Figura 3.15. Valor de la temperatura amb graus.

Al no disposar d'eines suficients per obtenir valors precisos, aquesta no és una mesura exacte. És una mesura que ens ajuda a fer-nos una idea del valor real de la temperatura i que en un futur caldria ajustar amb més precisió segons les necessitats del sistema.

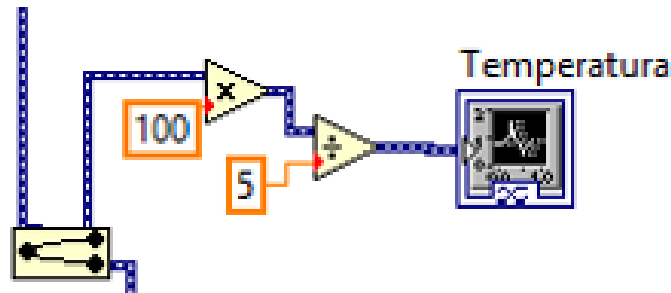


Figura 3.14. Diagrama blocs temperatura.

### Gràfic Pressió en BAR

El primer pas és fer una regla de tres tenint en compte que la sortida ens donarà un valor entre 0 – 5 Vdc, on 0 Vdc equival a 0 Bar, i 5 Vdc equival a 6 Bar.

Per fer una bona aproximació del valor real cal ajustar una mica els paràmetres ja que fer una regla de tres és insuficient. En aquest cas, s'ha restat 0,615 al valor obtingut de la regla de tres ja que el sensor ha estat calibrat per una entrada de 5 Vdc. No és la millor calibració fer falta d'eines però ens serveix per fer-nos una idea aproximada del valor que dona. Aquesta calibració es pot veure a la figura 3.16.

Si l'alimentació del sensor s'ajusta entre 9-32 V no farà falta restar-li els 0,615 en el cas del primer gràfic ("barometric" ajustat a Bar) i el segon 615 ("hPa" ajustat a hectopascals).

### Gràfic Pressió en hPA

Es passa les dades de pressió en Bar (abans de l'ajust de 0,615) a hPA sabent que 1 BAR = 1000 hPa, per tant:

$$hPa = Bar * 1000$$

Aquesta dada s'ha decidit mostrar ja que els hPa és la unitat que s'utilitza més habitualment per unitats de pressió.

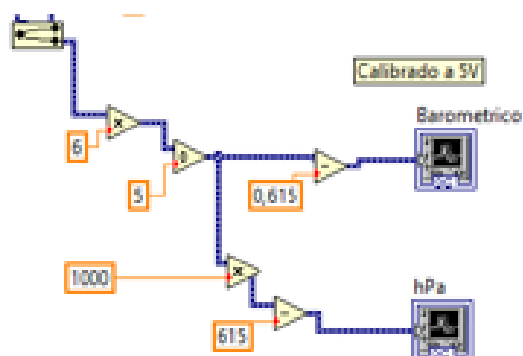


Figura 3.16. Diagrama blocs sensor pressió.

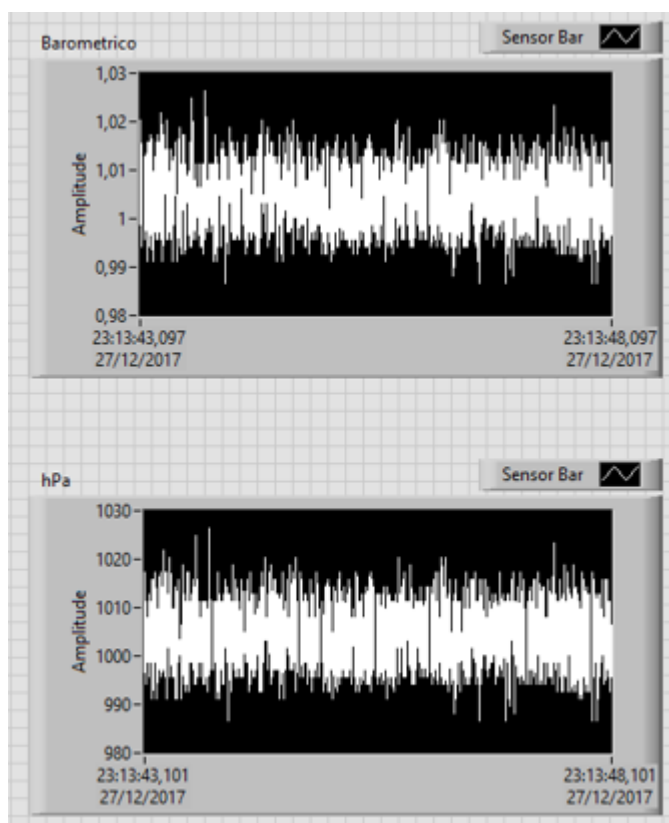
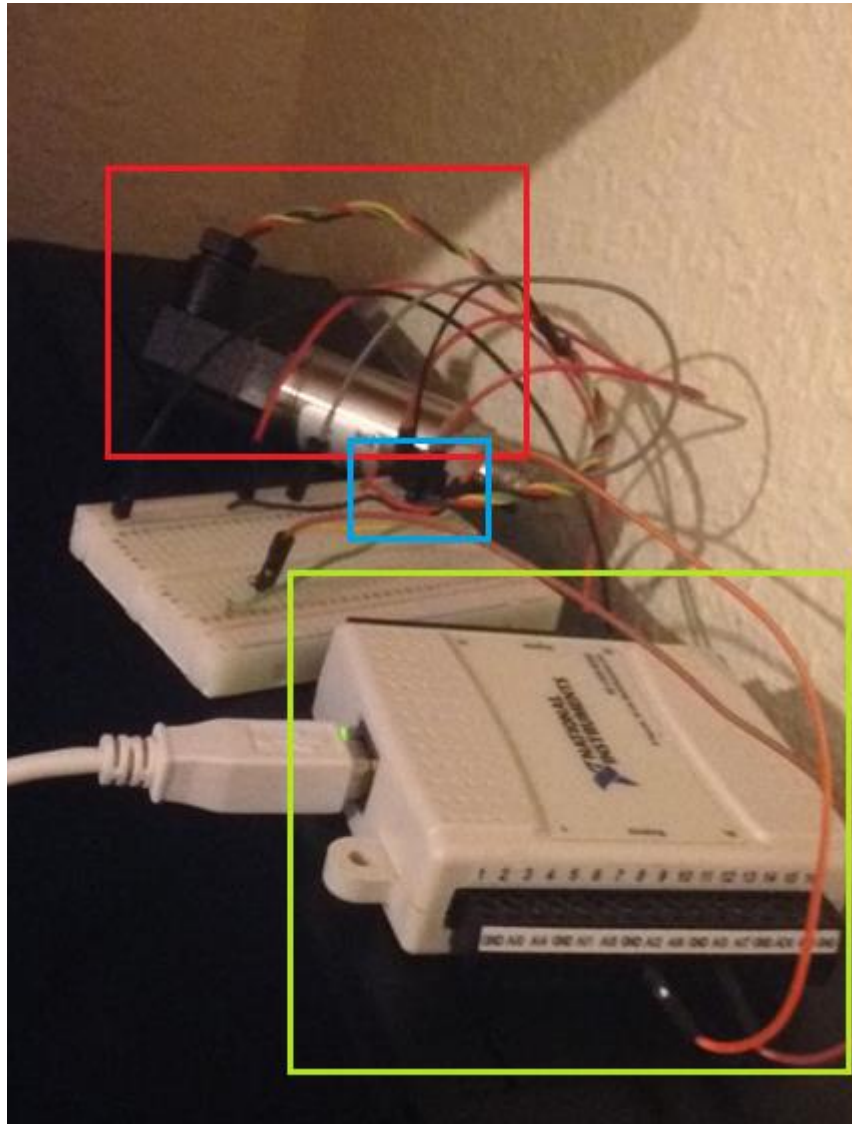


Figura 3.17. Valors de pressió en Bar (sobre) i hPa (sota).

Per finalitzar, es mostra una imatge del muntatge realitzat.

En la protoboard hi ha situat el sensor de temperatura LM35 i el sensor de pressió de RS Components.



*Figura 3.18. DAQ (verd) amb sensor de temperatura (blau) i pressió (vermell).*



### **3.4. Conclusions**

S'han assolit els objectius d'aquest capítol, que eren estudiar i implementar una nova unitat de control, en el cas d'emergència, substituïnt l'antic PC104+ per un miniPC Intel NUC5i7RYH sense que suposi una forta intrusió en l'antic sistema.

Tot i això, s'han de tenir en compte diverses consideracions d'alta importància a l'hora d'implementar aquesta nova unitat de control:

- 1) Alt consum.  
La NUC5i7 té un consum d'uns 28 W, molt superior al PC104+. Això, afectaria a la autonomia del Guanay.
- 2) Utilització d'un convertidor DC/DC entre 19-12 Vdc.  
La NUC pot funcionar entre 19-12 Vdc, per sobre o per sota d'aquest voltatge deixa de funcionar.
- 3) Desinstal·lació actualitzacions.  
És un requisit immediat a l'hora d'implementar dins del Guanay 2.1 (antic Guanay II), ja que sinó ens podríem emportar algun ensurt.
- 4) Posada en marxa del Labview.  
Un cop iniciat el sistema operatiu i en funcionament, el Labview hauria de funcionar immediatament.

La NUC pot aportar molts avantatges en un futur per un ampli processament de dades.

Ara bé, si s'implementa com a solució definitiva s'haurien d'utilitzar altres sistemes operatius com per exemple Linux, que és més configurable i permet tenir un control més complet dels processos del sistema operatiu.

També s'hauria de fer ús de plataformes lliures, ja que amb els preus de les llicències dels programes, el pressupost s'eleva considerablement.

## **CAPÍTOL 4. GUANAY 3.0**

### **4.1. Introducció**

En aquest capítol es planteja una nova arquitectura de la unitat de control degut a l'avanç de les tecnologies, a l'adaptació de les noves necessitats que sorgeixen en el submarí i als problemes que té l'actual estructura Guanay II.

### **4.2. Objectius**

Els objectius d'aquest capítol són la utilització de plataformes Open Source, programari i codi lliure, intercanvi amb altres universitats sobre coneixements, interconnectivitat amb altres AUV, arquitectura més rígida per solucionar els problemes que té l'actual model i una futura expansió de components i possibilitats.

Tots aquests objectius permetran que el Guanay 3.0 tingui més funcionalitat amb la computació i tractament de dades, facilitant el processat d'aquestes i del posterior anàlisi. També permet ampliar el nombre de dispositius dins el vehicle, i la capacitat per incloure sensors més moderns.

A més, es té en compte reduir els costos de manera que no perjudiqui la seguretat del vehicle. La relació entre costos i seguretat del sistema ha d'estar garantida.

En el CAPÍTOL 2 es detallen els programes principals que són lliures i es poden utilitzar per portar endavant aquest projecte. També s'afegeix el hardware utilitzat, no tot és lliure en aquest cas.

A continuació es mostren eines lliures que són d'utilitat:

- Disseny de PCB: Kicad, Eagle versió estudiant.
- Simulació de circuits: Orcad Demo
- Llenguatges: C/C++, Python, ....
- Hardware: Arduino, Raspberry, Beagle Bone, NUC.
- Sistemes operatius (SO): Linux, Free RTOS.

### 4.3. Nova Arquitectura Unitat de Control

#### 4.3.1. Guanay II

Aquest apartat ha estat explicat anteriorment al CAPÍTOL 3.

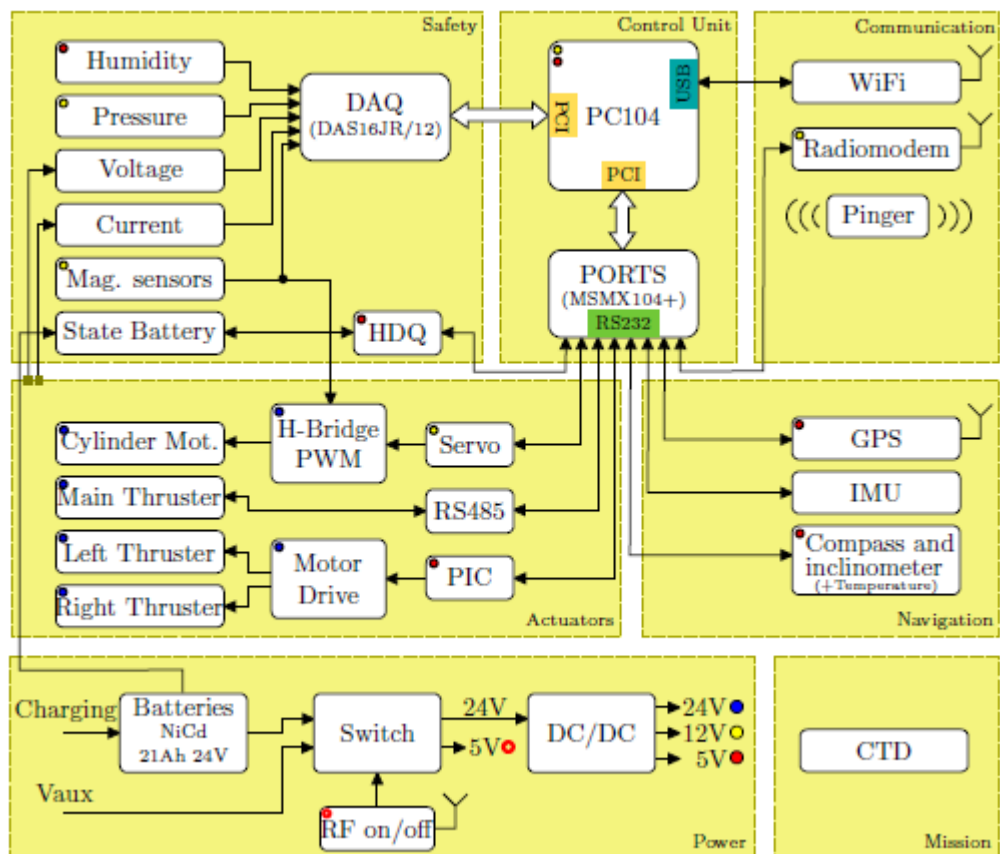


Figura 4.0. Arquitectura Guanay II [1].

No obstant això, per veure gràficament les diferències entre les dues arquitectures es mostra de nou l'arquitectura general del Guanay II.

Doncs bé, un cop vista l'arquitectura del Guanay II i veient els problemes que han anat sorgint, com la falta de seguretat en el sistema, s'ha decidit replantejar una nova distribució de la unitat de control.

Un dels problemes principals que té la unitat de control del Guanay II és que no disposa d'un sistema de seguretat que permeti saber l'estat de la unitat de control (PC104+) o bé un sistema que reinici aquesta unitat de control.

El segon problema i no menys important que el primer, és la poca capacitat de computació i emmagatzematge de dades [CAPÍTOL 2.]

A continuació s'explicarà la nova distribució de la unitat de control. No s'han entrat en detall amb els blocs de comunicacions, sistema d'energia, navegació, etc, així com el voltatge i potència en sensors i motors, ja que el que interessa és la nova forma que pren la nova arquitectura.

#### 4.3.2. Guanay 3.0

Aquesta idea ha sorgit degut a les noves tecnologies i als problemes de seguretat que té el Guanay II i de la limitació de sensors dels que disposa a l'interior del vehicle.

També se segueix amb la filosofia del projecte que són: estalvi de recursos econòmics, plataformes Open Source i codi obert per poder treballar amb altres universitats.

L'esquema que s'ha seguit per a la nova arquitectura es troba a la Figura 4.1. Aquest planteja una comunicació I2C i USB entre el components, en forma de malla.

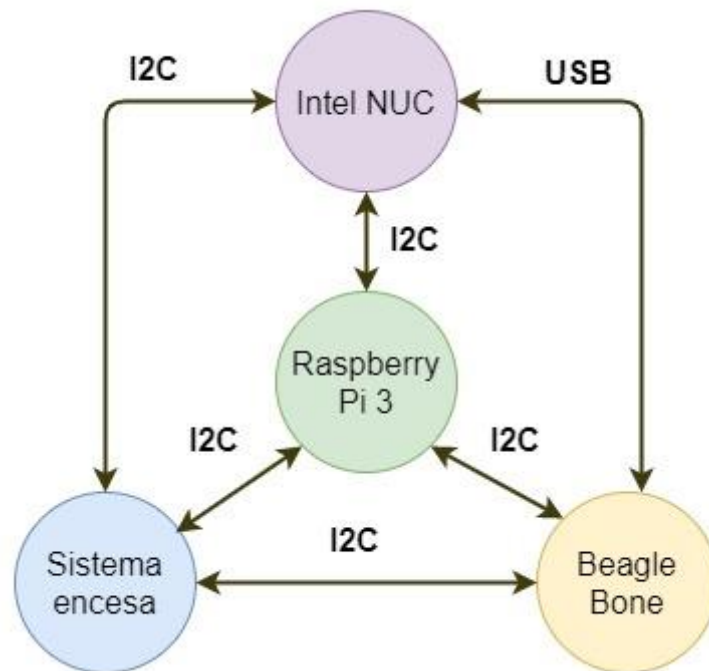


Figura 4.1. Esquema bàsic del Guanay 3.0.

Aquest és un sistema descentralitzat que disposa de suficients dispositius perquè el vehicle pugui tornar a la superfície o bé solucionar problemes interns, com el mal funcionament d'un dispositiu aplicant-li un reinici.

Compta d'un dispositiu de seguretat (Raspberry Pi 3), una unitat de control (NUC5i7), un sistema d'adquisició i tractament de dades (BeagleBone Black) i un sistema d'encesa.

Un cop queda definit aquest nou esquema per a la nova arquitectura, es planteja el sistema complet a la Figura 4.2.

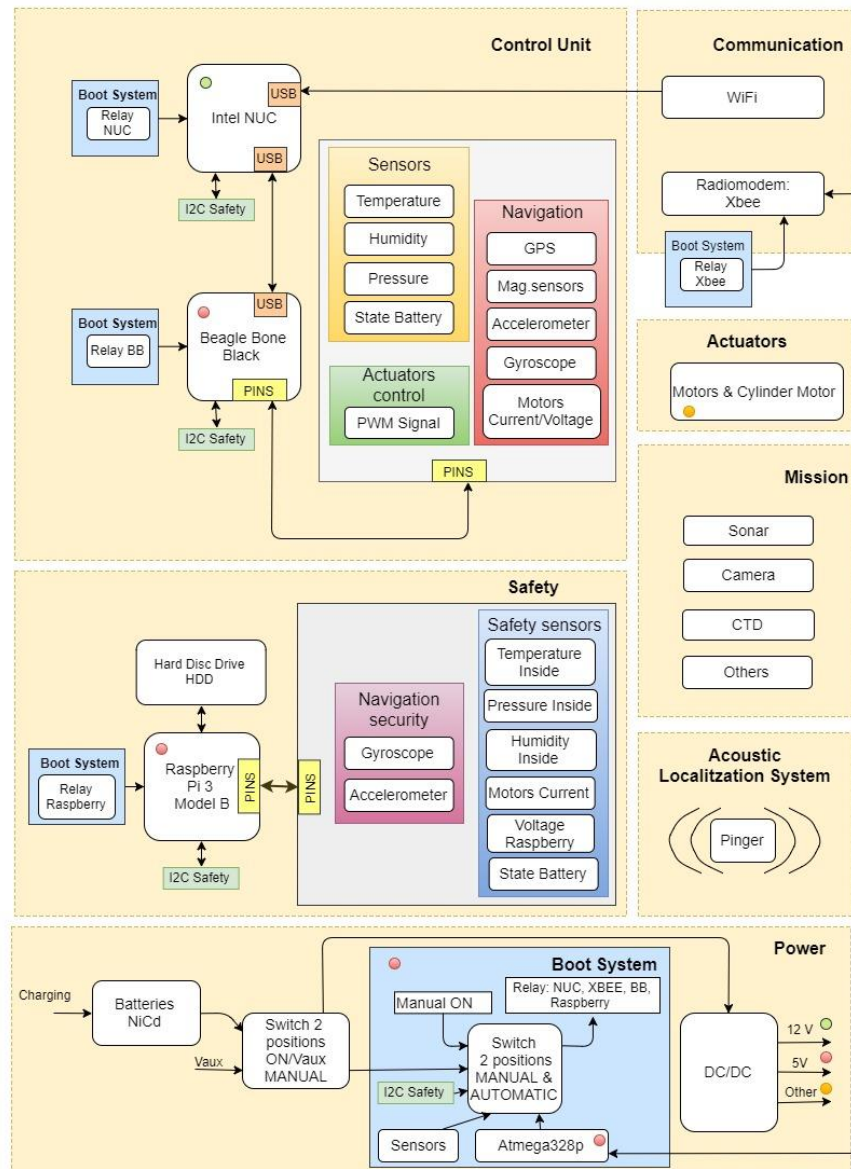


Figura 4.2. Arquitectura del Guanay 3.0.

Com es veu en la Figura 4.2, s'opta per la descentralització dels diferents dispositius, això permet un major control i gestió de tots els components i dispositius, alhora que dota al sistema de més seguretat. No obstant això, es torna un sistema més complicat en quant a software (programació).

A continuació s'expliquen els diferents blocs:

### **Control Unit**

La unitat de control principal està basada amb una Intel Nuc5i7RYH i permet tractar una gran quantitat de dades.

El sistema d'adquisició de dades dels sensors que trasllada i interacciona amb la unitat de control principal és la BeagleBone Black.

Aquests dos dispositius estaran connectats a través del port USB (NUC) i miniUSB (BeagleBone).



*Figura 4.3. Dispositiu Intel NUC [28]*



*Figura 4.4. Dispositiu BeagleBone Black [27].*

A nivell funcional formaran la unitat principal d'adquisició, tractament i emmagatzematge de les dades que proporcionen els diferents sensors del vehicle. Els sensors que analitza la BeagleBone poden ser sensors de seguretat com sensors externs acoplats al vehicle.

També, a través de la BeagleBone Black es controlen els actuadors, en concret els motors pel moviment del vehicle i el pistó que permet submergir o emergir el submarí. Aquests van controlats amb una senyal "Pulse Width Module" (PWM) que es transmet a través de diferents pins de la BeagleBone Black, cada pin amb el seu motor i pistó corresponent. Aquest sistema seria el mateix que es fa servir al Guanay II a partir d'un PIC. Per subministrar la potència als motors és necessari un pont en H.

## Safety

Sistema de seguretat basat en la Raspberry Pi 3 Model B.

Aquest sistema es basa en la comprovació constant de les comunicacions i l'estat dels diferents dispositius mitjançant la comunicació de seguretat I2C, així com l'adquisició de dades a través de diversos sensors que comprovin paràmetres de seguretat en el propi vehicle.



Figura 4.5. Safety: Raspberry Pi 3 Model B [24].

Aquest bus de comunicació (I2C) ens permet que la Raspberry es comuniqui amb la NUC, BeagleBone i el ATmega328P, aquest últim permet reiniciar qualsevol dels sistemes en cas d'errors greus o simplement per apagar-los.

El programa per fer la comprovació del estat dels dispositius mitjançant el bus I2C és de lliure elecció per al programador però es recomana que la Raspberry sigui el "Master" (cap) i els altres dispositius (NUC, BeagleBone i ATmega328p) siguin els "Slaves" (esclau). Així, el "Master" enviarà una pregunta a un "Slave" en concret, i aquest li respondrà amb una dada més complexa o un simple "OK". Fins que no rebi la dada esperada, el "Master" segueix preguntant al "Slave". Si passat un temps prudencial no s'efectua aquesta resposta per part del "Slave", el "Master" (Raspberry) segueix avaluant la resta del sistema i acabarà prenent una decisió sobre el que s'ha de fer. Per exemple: reiniciar el sistema fallit, emergir o una parada d'emergència total. Tot això dependrà del programa que implementi el programador d'aquest dispositiu.

Evidentment, si el sistema de seguretat falla, els altres dispositius se seguiran comunicant entre ells a través del mateix bus I2C, modificant el tipus de protocol (Master/Slave). En la Figura 4.5 es pot veure aquesta comunicació com "Data Support".

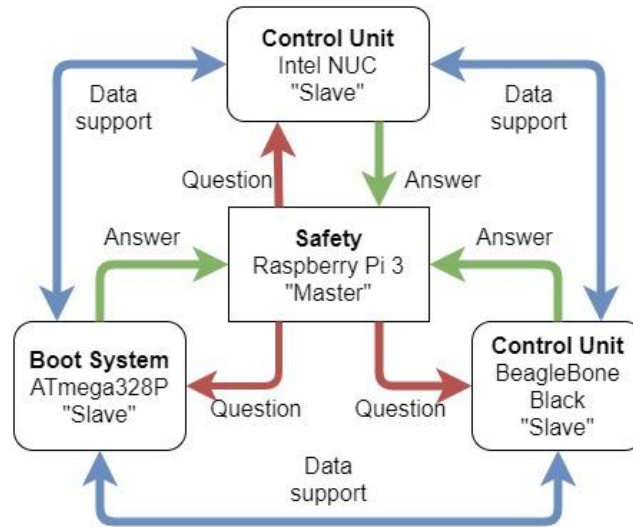


Figura 4.6. Funcionament del sistema de seguretat.

Les dades queden guardades en un disc dur extern que posteriorment es pot extreure per veure el comportament del sistema. Així, en el cas de falla general es pot veure on ha ocorregut.

L'encesa d'aquest dispositiu no és com la dels altres dispositius. Un cop el microcontrolador hagi encès aquest sistema (perquè li ha dit l'usuari), es queda sempre encès. No es pot apagar a no ser que el usuari tregui el Guanay del mar i el desmunti. Això s'aplica ja que al ser un sistema de seguretat quedarà fins l'últim moment encès, emmagatzemant totes les dades que tracta en el disc dur extern. De fet, es podria parlar d'una mena de caixa negra com als avions.

### **Boot system**

És un sistema d'arrencada basat en el microcontrolador ATmega 328P.

Constarà d'una comunicació per radiofreqüència (RF) a través d'un mòdul Xbee. Si bé aquest mòdul podria servir com a únic dispositiu per l'encesa del sistema, en aquest cas s'acopla amb el ATmega328P per permetre un major control.

Mitjançant una aplicació d'ordinador anomenada XCTU se l'hi podrà enviar diferents comandes al microcontrolador que permetran fer una encesa completa o parcial (a través d'uns relés) de tots els dispositius de control i de seguretat del sistema.



També ens permetrà obtenir entre d'altres dades, la temperatura i pressió interior del cilindre, estat de les bateries, estat de la comunicació I2C.

Aquest sistema s'explica amb més detall en el CAPÍTOL 5 ja que és un disseny propi.

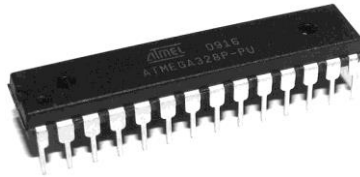


Figura 4.7. Boot system: ATmega328P-PU [19].

### **Acoustic Localization System**

Sistema ja inclòs en el Guanay II. És un element que emet senyals acústics i que ens permet saber la localització del submarí.

### **Mission**

Inclou el CTD, explicat anteriorment. Mesura dades com la salinitat i la temperatura de l'aigua.

### **Communication**

Aquest apartat inclou el WiFi, radiomòdem i Xbee (amb protocol de radiofreqüència).

En l'arquitectura es posen aquests tres sistemes.

El WiFi permet programar la NUC, Beagle Bone, Raspberry i ATmega328P.

El radiomòdem permet interactuar amb el software del sistema, i enviar/rebre dades.

### **Actuators**

Sistema que inclou els motors i cilindre per fer moure i submergir o emergir el Guanay. Aquest sistema no s'explica amb detall ja que requeriria d'un altre estudi més complet sobre aquest tema en concret.

### **Other**

Es fa un plantejament de nous dispositius que poden anar inclosos en el submarí, en aquest cas una càmera i un sónar.

Aquest últim hauria d'estar aïllat de tota la electrònica ja que provoca unes enormes interferències.

### **Safety bus**

La nova arquitectura inclou un bus I2C de seguretat (I2C Safety), que permet connectar-se als altres dispositius amb només dos cables d'informació, una alimentació i una massa.

En aquest bus (I2C) hi circula la informació entre els dispositius que permetrà saber el seu estat. Així doncs, la NUC, BeagleBone i el microcontrolador ATmega328P enviaran pel bus I2C informació al sistema de seguretat (Raspberry Pi 3). Aquest dispositiu guarda les dades en un disc dur alhora que avalua si tots els dispositius es troben correctament (no s'hagin quedat penjats o hi hagi algun error en la comunicació).

No obstant això, si la Raspberry falla, la pròpia NUC, BeagleBone i ATmega328P tenen sistemes propis de seguretat integrats mitjançant software (programació).

## **4.4. Funcionament de la nova arquitectura**

El funcionament de la nova arquitectura segueix els passos següents:

Mitjançant un ordinador portàtil o tablet amb una interfície gràfica i un Xbee, comunicats a través de USB, s'envia una comanda al Xbee situat al dispositiu d'encesa. Aquesta comanda és enviada al microcontrolador ATmega328P, que processa la informació i actua segons la dada enviada.

Això permet al usuari encendre, apagar, reprogramar entre d'altres accions a través del Xbee.

L'encesa del dispositiu pot ser de dues maneres:

1. Encesa completa ("ON"): Encén tots els dispositius. Un cop el sistema fa les seves comprovacions via software, revisant que tots els dispositius funcionen, el vehicle pot emprendre la missió.  
El sistema de seguretat i els altres dispositius utilitzen eines de software per controlar el bon funcionament de tots els dispositius.  
En tot moment hi ha disponible una comunicació tipus Wifi i de radiofreqüència que permet visualitzar les dades "en directe" a partir del portàtil i una interfície gràfica, també permet actuar sobre el vehicle.
2. Encesa dels dispositius programables ("OFF"): Encén la NUC i la BeagleBone, i permet via Wifi reprogramar aquests dos dispositius o modificar-ne uns paràmetres determinats. En el cas que es vulgui començar una missió s'haurà d'enviar una comanda per a l'encesa completa del vehicle.

#### 4.5. Consum potència

Es tindrà en compte els dispositius electrònics de control, així doncs, no es tindràn en compte els motors, pistó (per submergir i emergir el submarí) i d'altres components ja que en una nova versió del Guanay 3.0 s'haurien d'escollir uns components més eficaços i de menys consum.

El consum de la nova arquitectura (unitat de control) és molt superior a l'antiga del Guanay II, ja que com s'ha vist anteriorment, només comptava amb el PC104+ com a unitat de control i un microcontrolador PIC i DAQ com a sistemes per al control del vehicle.

A continuació es mostra una taula dels valors màxims aproximats del consum dels dispositius per al control del vehicle Guanay II.

Sistema	Potència
U.C: PC104+	6 W
A.D: DAS16JR/12	2,3 W
C.M: PIC16F1508	2,5 W
TOTAL	10,8 W

Taula 4.0. Potència Guanay II.

U.C = Unitat de Control.

A.D = Adquisició de dades.

C.M = Control Motors

A continuació es mostra una taula dels valors màxims aproximats del consum de la nova unitat de control, sense comptar amb els sensors i comunicacions.

Sistema	Potència
U.C.P: Intel NUC5i7RYH	28 W
S.E : Placa I+D+I	5,4 W
S.S: Raspberry Pi 3 Model B	12,5 W
C.A.D: BeagleBone Black	2,3 W
<b>TOTAL</b>	<b>48,2 W</b>

Taula 4.1. Potència Guanay 3.0 .

U.C.P = Unitat de Control Principal

S.E = Sistema d'Engegada.

S.S = Sistema de Seguretat.

#### 4.6. Justificació de la nova arquitectura

Aquesta nova arquitectura s'ha pensat tenint en compte diversos factors.

S'ha seguit la filosofia Open Source, codi lliure, plataformes lliures, i gran comunitat amb una gran informació disponible, a part de buscar altres dispositius econòmicament viables.

Tots els components amb els que s'ha dissenyat l'arquitectura són altament fiables.

Aquesta arquitectura néix amb la idea profunda de solucionar tots els problemes de seguretat que no es tenen en compte en el Guanay II i que poden posar en risc la seva integritat.

També incorpora una novetat força important respecte amb l'anterior projecte (Guanay II) i és que tots els dispositius (NUC, BeagleBone, ATmega328P i Raspberry) poden ser programats a través de Wifi i del bus I2C. Tot això, permet tenir l'estructura mecànica montada sense haver de desmontar cap component.

S'ha utilitzat un protocol de comunicació I2C [49], que permet en aquest cas, moure les dades de seguretat (bàsicament), ja que és un protocol amb forma de bus serie que ens permet tenir fins a 256 dispositius connectats, és a dir, que a part de la NUC, BeagleBone, Raspberry i ATmega328P poden haver-hi 252 dispositius més connectats al mateix bus, on cadascun d'ells, té una adreça concreta. En aquest bus s'hi podrien incloure sensors, actuadors, microprocessadors i controladors, entre d'altres.

Aquest bus té quatre cables: Data(SDA), Clock (CLK/SCL), Alimentació (Vcc) i Massa (GND).

Per evitar interferències i altres problemes de pèrdues de dades s'ha de posar una resistència de pull-up entre SDA i Vcc, i entre CLK i Vcc.

És un protocol força robust a curta distància però a partir de 2 metres es comença a perdre el senyal. També influeixen molt les interferències electromagnètiques (EMI), per tant, s'ha d'aïllar (apantallar) correctament.

No obstant, s'ha decidit utilitzar aquest protocol de comunicació per la alta simplicitat que té.

A continuació es mostra la comunicació del bus I2C en variis dispositius.

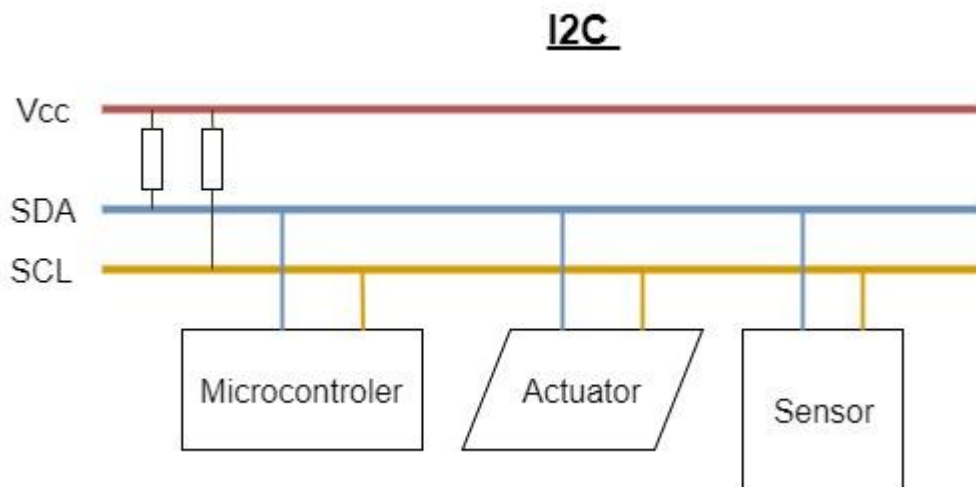


Figura 4.8. Bus I2C [49] .

#### 4.7. Conclusions

Aquest nou sistema solventa desavantatges que té l'arquitectura del Guanay II. Injecta sistemes de seguretat que permeten que el Guanay compti de sistemes de software i hardware de control i fer d'ell un vehicle més autònom. A més, tots els dispositius són actuals i per tant, amb més capacitat per gestionar i emmagatzemar dades.

S'opta per un model de descentralització de les unitats de control, ja que així, en cas de fallada, el propi sistema pot autogestionar-se, és per això que necessita un sistema de software potent i robust.

Ara bé, el control que pot obtenir el Guanay amb aquesta nova arquitectura és molt superior a la anterior.

No obstant, el nou model quadruplica el consum dels sistemes de control. Això és un factor que cal tenir en compte i s'hauria de solventar amb un nou sistema de bateries.

Cal dir que no se sap l'estat actual del submarí i per tant s'ha replantejat amb l'últim document que hi ha actualitzat. Això és degut a que es fan força treballs de final de grau, màster i doctorat amb aquest vehicle. Per tant, és complicat recopilar tota la informació per analitzar-la. Vist aquest problema i amb l'objectiu de dur a terme aquest projecte, fóra bo fer un llistat de novetats i treballs que es fan sobre el vehicle.

Per finalitzar, en tot vehicle (concretament en submarins) es pot allargar el seu cicle de vida fent una revisió de mitja vida, canviant els components desgastats (bateries) i electrònica. Aquest projecte (Guanay 3.0) és un bon candidat per fer aquesta renovació.

#### **4.8. Nous horitzons**

Aquesta nova arquitectura ens permet ampliar nous horitzons amb els AUV i dels diferents camps d'investigació dins de la universitat.

Degut a l'alt processament de dades que pot realitzar la unitat de control principal Intel NUC5i7RYH podem integrar nous sistemes d'intel·ligència artificial, visió per computador, alta robustesa del sistema, noves rutes d'exploració, radars, sónars i d'altres.

S'ha de tenir en compte que si s'inclou un sónar pot afectar greument a la electrònica degut a les interferències (EMI) que aquest genera, per això, faria falta aïllar l'electrònica del sónar.

També amb els nous sistemes de comunicació es pot enviar una quantitat més extensa de dades per poder analitzar i tractar externament, i per exemple, fer un ús d'impresores 3D per fer impresions de peces sota l'oceà.

Per finalitzar, un ús molt interessant, que en un futur tindrà un ús molt extens, és la descentralització de les funcions (missions) dels AUV. És a dir, comptar amb més unitats de AUV on cada unitat està especialitzada en una tasca però alhora, tindrà altres AUV amb els que realitzar tasques col·laboratives. Les dades poden ser enviades d'un a l'altre per fer una computació més elevada de les dades.

## **CAPÍTOL 5. DISSENY DEL SISTEMA D'ENCESA PER AL GUANAY 3.0**

Aquest capítol tracta de la investigació, desenvolupament i innovació d'un disseny electrònic per a l'encesa dels diferents dispositius del Guanay 3.0.

El Guanay 3.0, com bé s'ha explicat en el CAPÍTOL 4, conté una nova arquitectura pel funcionament del vehicle. Això implica nous dispositius com la Intel NUC, BeagleBone Black que s'encarreguen de la unitat de control i adquisició de dades, i la Raspberry que s'encarrega del sistema de seguretat.

En aquest punt, prenem força la idea de crear un sistema d'encesa, ja que un cop tancada la càpsula (cilindre) on està tota l'electrònica, és necessari posar en marxa tots els dispositius.

Per tant, el capítol planteja un prototip per realitzar aquesta tasca, l'encesa dels diferents dispositius, tenint en compte diferents restriccions tant en software com en hardware.

Les restriccions que es tenen en compte per realitzar aquest disseny són:

- Emular els interruptors que permeten a l'usuari encendre qualsevol dispositiu.
- Comunicació remota mitjançant un dispositiu situat dins el vehicle.
- Enviament i rebuda de dades.
- Ús de plataformes Open Source [50] [51] [53] [54].
- Arquitectura robusta en software i hardware.
- Disseny i desenvolupament en PCB.

Per complir aquests objectius s'ha realitzat un disseny que compta amb un microcontrolador ATmega328P-PU, explicat en el Capítol 2.

Així doncs, per realitzar aquest dispositiu d'encesa el treball se centra en dos prototips, el prototip I (circuit bàsic), i el prototip II (disseny final del circuit d'encesa).

### **5.1. Protip I. Circuit bàsic amb el microcontrolador ATmega328P**

En aquest apartat s'ha realitzat el disseny i desenvolupament en placa de baquelita ("topos") un circuit que verifiqués els components mínims pel funcionament del microcontrolador que aniria al disseny final del prototip d'encesa. També es comprova el sistema de càrrega d'arxius dins el microcontrolador, instal·lant el bootloader, i posteriorment programes per comprovar que tots els pins del microcontrolador funcionaven correctament.

Per fer el disseny funcionament mínim (Figura 5.0) del microcontrolador es cerca informació en el datasheet (manual del fabricant pel model de microcontrolador) del ATmega328P, i sobre l'esquema i arquitectura del Arduino Uno, ja que aquesta placa de desenvolupament conté aquest microcontrolador. L'esquema del Arduino Uno està inclòs en l'Annex com a pdf.

Amb el programa Fritzing [54] (Figura 5.1) es realitza un muntatge de com ha d'anar el microcontrolador connectat per al seu funcionament bàsic, tenint en compte els pins del microcontrolador. En les següents figures es poden veure la disposició de pins i components.

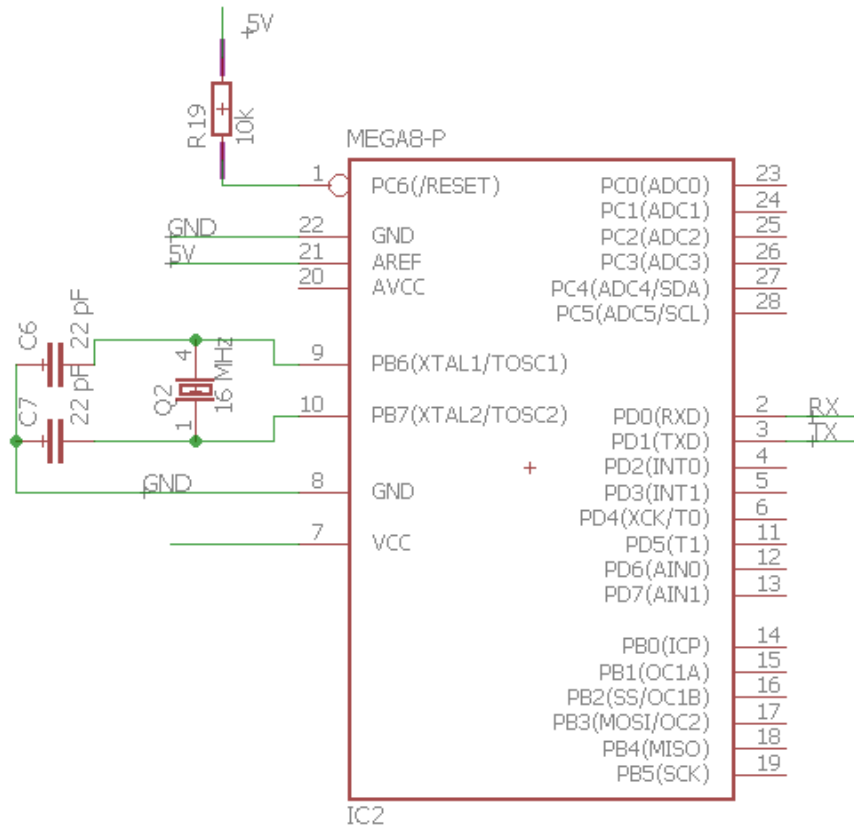


Figura 5.0. Esquema pel funcionament bàsic del prototip I.

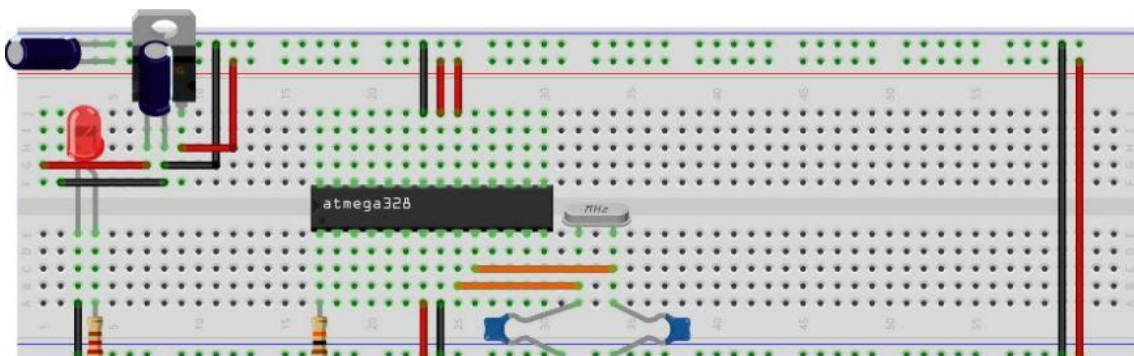


Figura 5.1. Prototip II en protoboard.



A la Figura 5.1 es pot veure l'esquema bàsic per al funcionament del microcontrolador. Els elements necessaris són:

- 1x Microcontrolador ATmega328P-PU
- 1x Cristall 16 MHz
- 1x Conversor linial DC/DC LM7805. (opcional pero s'afegeix per una linialitat en l'alimentació)
- 1x Resistència 10.000  $\Omega$
- 1x Resistència 220  $\Omega$
- 2x Condensador 10  $\mu$ F
- 2x Condensador 22 pF
- Cablejat

Un cop muntat el circuit, es programa el microcontrolador, per això és necessari un conversor sèrie de USB a TTL. En ambdós prototips s'utilitza un model FT232RL. Aquest conversor permet ser utilitzat per a microcontroladors de 3,3 V i 5,5 V. En el cas dels prototips es posa a 5 V.



Figura 5.2. USB a TTL model FT232RL.

El microcontrolador ATmega328P es programa per un protocol sèrie. Per fer-ho s'utilitzen els pins 2 i 3 del microcontrolador. Així doncs, utilitzant el conversor USB-TTL anomenat anteriorment es programarà mitjançant els pins RX i TX d'aquest. En concret, el pin RX del FT232RL es connecta al pin 3 (TX) del ATmega328P, i el pin TX del FT232RL es connecta al pin 2 (RX) del ATmega328P. L'alimentació a 5 V (Vcc) i la massa (GND) serà comuna per als dos dispositius.

Per realitzar la programació del microcontrolador s'utilitza la IDE (interfície) d'Arduino [50] que és Open Source i ens permet carregar fàcilment els programes. El funcionament per programar-lo s'explica a continuació.

El port del conversor és detectat per la plataforma sense cap mena de problemes, aquest agafa un nom tipus COM seguit d'un número, i a continuació es selecciona el microcontrolador com si fós la placa de desenvolupament Arduino Uno, així és com teòricament es carrega el programa. En la següent es pot veure com queda.

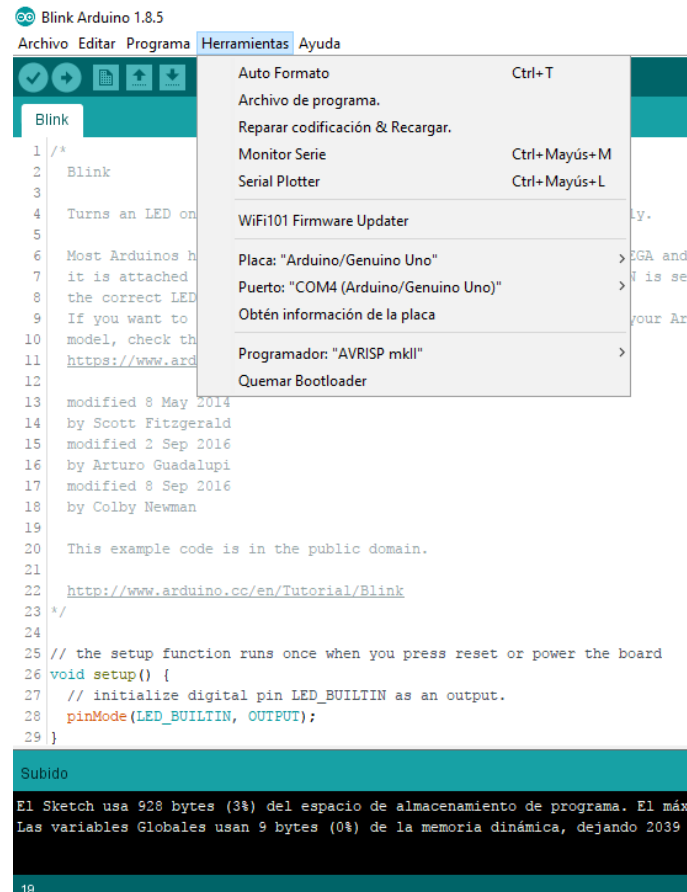


Figura 5.3. Selecció microcontrolador i port (COM).

Ara bé, després de moltes comprovacions del circuit de la Figura 5.0 i Figura 5.1 s'arriba a la conclusió de que no és possible programar el microcontrolador ATmega328P-PU, ja que la placa dissenyada només permet el funcionament bàsic del microcontrolador, sense la possibilitat de ser programat. Les proves que es realitzen es fan amb un ATmega328P extret d'una placa d'Arduino Uno, que contenia el bootloader programat i un altre microcontrolador ATmega328P verge, sense el bootloader.

El bootloader és el programa que permet llegir les dades enviades a través d'un convertidor sèrie USB a TTL i reescriure-les dins el microcontrolador. En el cas del ATmega328P va inclòs dins la Flash del microcontrolador, això pot ser un benefici (no es necessiten programadors externs) o un inconvenient (espai ocupat dins el microcontrolador). Per fer la càrrega del bootloader s'ha seguit manuals penjats a la mateixa web d'Arduino. A l'Annex I hi ha una imatge que correspon amb el connexionat necessari per a realitzar la càrrega del bootloader. Es requereix d'un altre microcontrolador amb el bootloader precarregat per poder realitzar aquesta tasca.

Així doncs, després de revisar el datasheet del ATmega328P, diferents foros i webs s'arriba a una solució. Aquesta passa per instal·lar un pulsador al pin de reset, que permet posar el pin en un estat "Low" i provocant un flanc de pujada "High" quan es deixa anar alhora que es transmeten les dades de programació a través del F232RL.

Això permet que es transfereixin les dades del programa al microcontrolador aconseguint una reprogramació del dispositiu. També s'aconsegueix programar el bootloader en el cas dels microcontroladors verge.

En la següent figura es pot veure l'esquema amb el pulsador de reset ja inclòs, que estarà normalment obert i amb un led amb la seva corresponent resistència d'aproximadament 200  $\Omega$  en el pin digital 13 (D13).

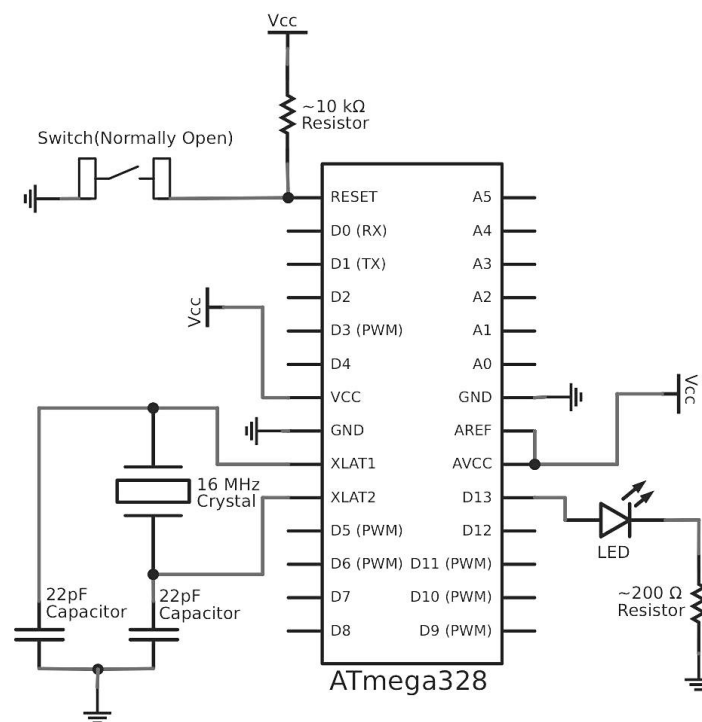


Figura 5.4. Esquema ATmega328P-PU amb pulsador.

L'alimentació (Vcc) segueix sent la mateixa que en la Figura 5.1 que passa per un convertidor LM7805.

Per carregar el programa cal mantenir el pulsador de reset apretat mentre es compila el programa al microcontrolador a través de la IDE d'Arduino.



Figura 5.5. Compilació programa.

Un cop s'hagi completat la barra de la dreta de la Figura 5.5, es deixa de prémer el pulsador.

```
28 | pinMode(LED_BUILTIN, OUTPUT);  
29 |  
Subiendo...  
El Sketch usa 928 bytes (2%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.  
Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las variables locales. El máximo es 2048 bytes.  
13 Arduino/Genuino Uno en COM4
```

Figura 5.6. Pujada programa.

A continuació ens sortirà un missatge a l'esquerra dient que s'està pujant el programa, seguit d'un altre missatge que ens diu que el programa ja està pujat.

```
27 | // initialize digital pin LED_BUILTIN as an output.  
28 | pinMode(LED_BUILTIN, OUTPUT);  
29 |  
Subido  
El Sketch usa 928 bytes (2%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.  
Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2039 bytes para las variables locales. El máximo es 2048 bytes.  
13 Arduino/Genuino Uno en COM4
```

Figura 5.7. Programa pujat.

Finalment, amb el circuit de la Figura 5.4 s'aconsegueix reprogramar el microcontrolador i instal·lar el bootloader en el cas del microcontrolador verge.

### 5.1.1. Altres problemes

S'han trobat altres problemes que han endarrerit aquest prototip, i per tant, el disseny final, ja que per fer el disseny final (prototip II) era requisit el funcionament bàsic del microcontrolador, que es fa en el prototip I. A continuació s'expliquen aquests problemes:

- Impossibilitat de programar el microcontrolador amb el circuit bàsic mitjançant el conversor FT232RL sense el pulsador de reset al microcontrolador.  
El conversor FT232RL compta amb un pin de sortida de reset, si aquest es connecta al pin de reset del microcontrolador fa la funció que malauradament ha de fer l'usuari prement el pulsador.  
Es desconeix el motiu per la qual cosa no funciona correctament el pin de reset.  
La solució finalment passa per afegir el pulsador (Figura 5.4) de reset al microcontrolador, cosa que a més, permet reiniciar el microcontrolador en cas que es quedi bloquejat.

- La utilització del microcontrolador ATmega328-PU i no el ATmega328P-PU. La diferència entre ambdós models és que el primer no compta amb la funció de “Picopower” i el segon sí. Aquesta funció ens permet programar el microcontrolador perquè tingui un baix consum.

El microcontrolador escollit va ser el ATmega328P-PU precisament per poder comptar amb aquesta funció ja que és d'utilitat en un disseny d'encesa, que només espera rebre comandes cada cert temps.

Però, a l'hora de fer la compra de components el fabricant va subministrat el model equivocat.

Després de fer una cerca intensa es va arribar a la conclusió que per fer anar aquest microcontrolador era necessari canviar el codi font del arxiu Arduino Uno/Genuino i posar una nova direcció (la del ATmega328-PU). Aquesta direcció es pot trobar al datasheet del fabricant. No obstant això, es va decidir descartar aquesta opció ja que era massa costosa i es va decidir comprar un ATmega328P-PU, que és amb el que es treballa en aquest projecte.

Si més no, es va trobar la solució al problema però, va requerir de moltes hores i dedicació ja que també depenia d'altres problemes com el del convertidor FT232RL.

## 5.2. Protip II. Disseny final del circuit d'encesa

Per a la realització d'aquest apartat, s'ha treballat amb les eines gratuïtes següents:

- Simulació de circuits: Orcad Demo.
- Desenvolupament a PCB: Eagle versió gratuïta per estudiant.
- Programació: IDE d'Arduino.

En un principi es va voler utilitzar l'eina Kicad per al desenvolupament en PCB, ja que és una eina completament Open Source, però degut al temps que es necessitava per dominar aquest programa es va decidir treballar amb Eagle versió gratuïta d'estudiant, ja que es tenia un coneixement i un domini previ.

També es volia portar la placa PCB a fabricació però degut a la falta de recursos i facilitats econòmiques de la UPC no va ser possible. Així doncs, es va fer el prototip II en una PCB de doble cara de fibra de vidre, on tot el procés de impremta es va dur a terme en el taller de la universitat.

Per fer el disseny d'encesa i emular el dit humà amb el polsador, s'ha pensat en un relé que activarà la bobina mitjançant unes comandes del microcontrolador, i la placa del relé que actua com un interruptor, pontreja l'interruptor de la Intel NUC, BeagleBone Black, Raspberry Pi i Xbee.

No obstant això, la Raspberry Pi no té cap interruptor/polsadores d'encesa, un cop s'alimenta s'encén automàticament, és per això que es proposa un circuit d'enclavament. Mitjançant un pin del ATmega328P s'activa un senyal que permet el pas de corrent a un transistor, així, la Raspberry

quedaria alimentada. En aquest moment, un pin de l'alimentació de la Raspberry anirà directament al transistor anterior, i mantindrà un voltatge i intensitat constant, amb la qual cosa, per més senyals que enviés posteriorment el ATmega328P (a través del seu pin) no actuaria sobre el transistor. També seria bo posar un díode perquè no circuli el corrent cap al microcontrolador. Això permet que la Raspberry que actua com a sistema de seguretat no es pugui apagar a no ser que l'usuari la desconnecti. Aquesta part no s'ha pogut realitzar ja que era preferible estar més centrat amb el software i el hardware general. En tot cas, seria una millora futura que s'hauria d'aplicar. A continuació es mostra una imatge de com hauria de ser aquest petit sistema d'enclavament, tot i que queda obert a un nou estudi.

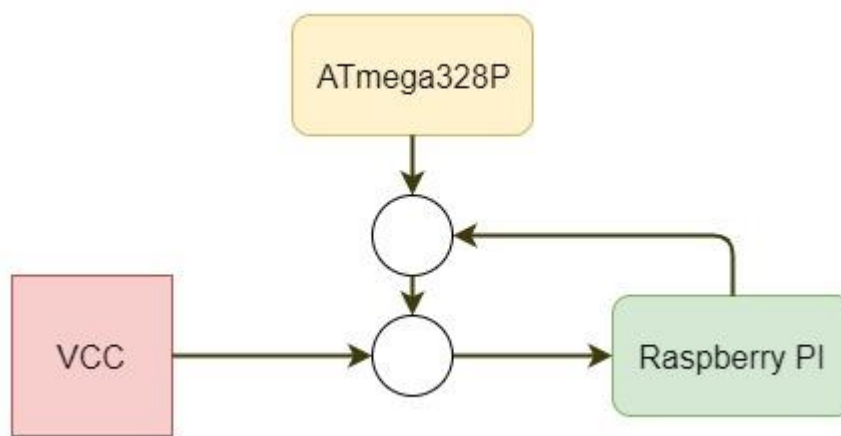


Figura 5.8. Circuit enclavament Raspberry.

Tot el hardware que permet la programació i funcionament del microcontrolador ATmega328P-PU és el mateix que el del prototip I però amb diferent col·locació.

### 5.2.1. Descripció de pins

Per a fer el hardware del circuit s'ha tingut en compte els diferents pins del ATmega328P.

En tres dels pins digitals s'hi han posat uns transistors per poder encendre els relés, ja que el propi microcontrolador no pot entregar una intensitat suficient per poder encendre'ls. Aquests relés van dedicats a la NUC, Beagle Bone i Xbee, on el del Xbee actua com un reinici, és a dir, que sempre estarà activat fins que l'usuari li enviï una comanda demanant el contrari.

Hi ha un altre pin digital dedicat al circuit d'enclavament de la Raspberry però, al no desenvolupar aquesta part de hardware es mostra amb un led.

En un altres tres pins digitals es dota el sistema d'uns leds (verd, groc, vermell) que permeten saber l'estat del sistema de comunicacions, o bé si hi ha encès algun relé.

Es deixen lliures els pins analògics per poder dotar el sistema d'un fotoresistor (LDR) i un altre pin per un sensor (LM35) que permet calcular la temperatura del mateix sistema.

A continuació es mostra la Taula 5.0 de com queden els pins del ATmega328P-PU en relació amb el hardware. On la inicial D indica que el pin és digital, i la inicial A indica que el pin és analògic.

El nom del pin és el mateix que el de la Figura 5.0.

<b>Numero pin ATmega328P-PU</b>	<b>Pins ATmega328P-PU</b>	<b>Hardware</b>
2	D0	RX per programar
3	D1	TX per programar
4	D2	RX per comunicació Xbee
5	D3	TX per comunicació Xbee
6	D4	Pin reinici del Xbee
11	D5	Led vermell per estats
12	D6	Led verd per estats
13	D7	Led groc per estats
14	D8	Sensor digital
15	D9	Pin encesa relé Intel NUC
16	D10	Pin encesa relé BeagleBone
17	D11	Pin per encesa Raspberry
18	D12	-

19	D13	-
23	A0	Fotoresistència (LDR)
24	A1	-
25	A2	-
26	A3	Sensor de temperatura
27	A4	-
28	A5	-

Taula 5.0. Pins del prototip II. Disseny final del sistema d'encesa.

### 5.2.2. Control manual i automàtic

La placa compta amb tres interruptors que els hi corresponen dos pulsadors. Aquests interruptors ens permetran encendre de manera automàtica (via software) o bé manual, els diferents relés. El interruptor canviarà el mode manual/automàtic, i els pulsadors permetran encendre els relés.

Els tres interruptors corresponen a:

- Relé del Xbee amb el suport del pulsador per reiniciar manualment.
- Relé Intel NUC amb el suport d'un pulsador (comú amb la Beagle Bone).
- Relé BeagleBone Black amb el suport d'un pulsador (comú amb la Intel NUC).

### 5.2.3. Sistema d'encesa software

El sistema d'encesa per software requereix d'una robustesa, ja que en el cas d'enviar varies comandes del ordinador al xbee, no pot ésser que el microcontrolador actui sense tenir un ordre. O bé que repeteixi comandes. Per exemple:

Si l'usuari envia "ON" mitjançant el ordinador, el submarí fa la seqüència perquè els relés engegin tots els dispositius. En el cas de que l'usuari torni a enviar un altre "ON", el microcontrolador no pot



fer la mateixa seqüència ja que sinó s'apagarien tots els dispositius. Cal recordar que els relés actuen com el dit humà quan pulsa el botó d'encesa d'un dispositiu.

Així doncs, la Figura 5.9 mostra el diagrama d'estats creat perquè no succeeixin aquests casos, a més, s'adjunta el codi del seu funcionament:

- 'x,y,z,w': són variables de control.
- function(a), function(b) i function(c): són les funcions que realitzarà cada apartat.
- lectura: és la instrucció que s'envia a través del ordinador mitjançant el Xbee.

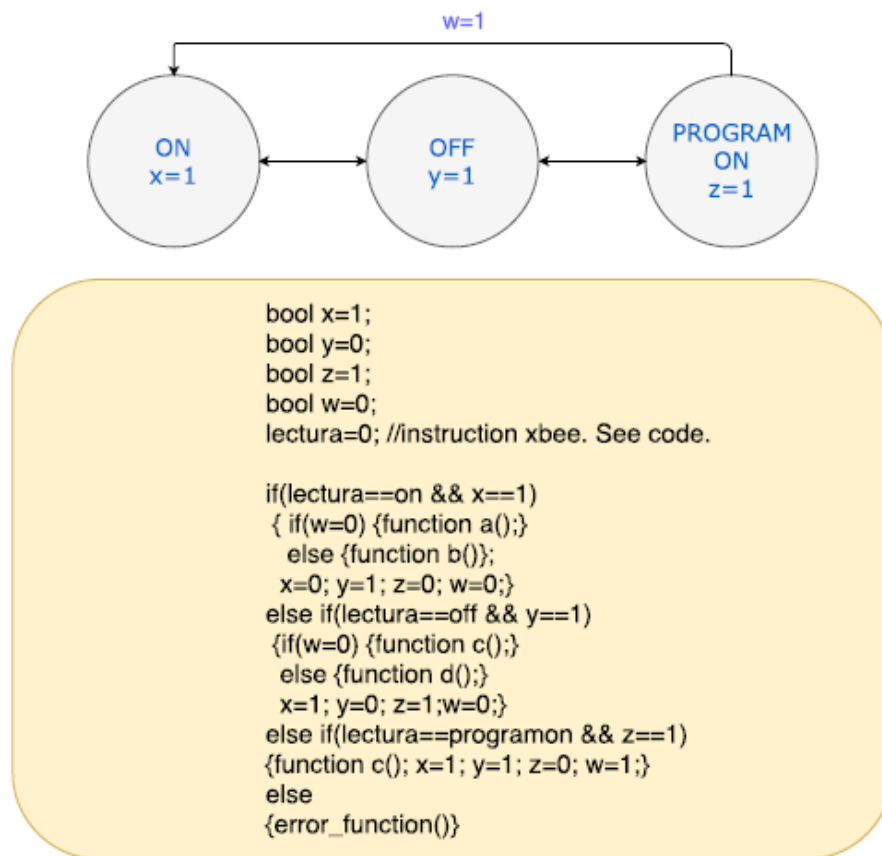


Figura 5.9. Diagrama d'estats del sistema d'encesa.

Inicialment el sistema es troba en "OFF", per tant, només es poden seleccionar els modes: "PROGRAM ON" i "PROGRAM OFF".

La diferència entre els aquests modes són:

- **"OFF"** : Apagada dels dispositius que estan en funcionament.

- **"ON"** : Encesa de tots els dispositius del sistema, la Intel NUC, Beagle Bone i la Raspberry, on aquest últim es recorda que és el sistema de seguretat del vehicle i que un cop encès ja no es podrà aturar via software.
- **"PROGRAM ON"** : Només encén els dispositius que es poden reprogramar, un cop està tota la arquitectura montada, que són la Intel NUC i la Beagle Bone.

El ATmega328P-PU estarà programat perquè estigui en mode "Sleep" i només actui quan reb una dada a través del Xbee (connectat als pins 2 i 3) o bé a través de la comunicació I2C que es té amb altres dispositius. Així evitem un ús innecesari del microcontrolador i podrem evitar el consum quan estigui apagat el submarí, ja que si es deixa tot connectat, el microcontrolador ATmega328P-PU tindrà un consum, a l'espera de les comandes del Xbee.

El mode "Sleep" ens permet programar el microcontrolador amb un model de baix consum deixant al microcontrolador "dormir" a la espera de rebre alguna senyal. Pasa de consumir uns 60 mA a uns 15 mA, i inclús si es programa el propi Xbee amb aquest mode "Sleep", el circuit passa a consumir uns 5 mA. També es pot tenir apagat el microcontrolador ATmega328P-PU i que el Xbee s'encarregui de gestionar les dades rebudes, tot això amb el seu propi mode "Sleep", amb un consum d'uns 10 uA. Aquest mode "Sleep" no s'ha pogut implementar ni al microcontrolador ATmega328P-Pu ni al Xbee per falta de temps.

#### 5.2.4. Comandes registrades

El programa que s'ha creat disposa de les següents comandes:

**"ON"** = 'i'

**"OFF"** = 'o'

**"PROGRAM ON"** = 'p'

**"REINICIO XBEE"** = 'r'

**"TEMPERATURA"** = 't'

Les variables de l'esquerra indiquen el nom de la variable dins el programa, i la de dreta és la variable que envia l'usuari a través del ordinador.

### 5.2.5. Xbee

És un petit dispositiu de la marca Digi [55] que inclou un microcontrolador amb un mòdul de radiofreqüència.



Figura 5.10. Xbee S2C [56].

El model escollit per aquest treball final de grau és el Xbee S2C generació A.

És el model més econòmic, uns 20 €, i disposa de llibreries ja fetes per el microcontrolador ATmega328P.

El seu mòdul de radiofreqüència (RF) treballa a 2,4 GHz i pot arribar a una distància d' 1 km. Hi ha altres models que arriben fins a 10 km de distància.

Xbee fa servir una comunicació en forma d'estrella. A la Figura 5.11 es pot veure un exemple.

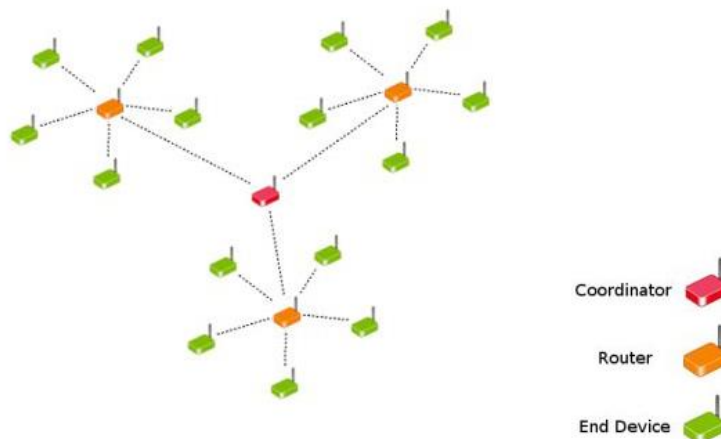


Figura 5.11. Xarxa amb forma d'estrella.

La Figura 5.11 mostra el que seria el coordinador principal ("Master") en vermell i els esclaus ("Slave") en taronja i verd.

Aquesta comunicació amb forma d'estrella és típica de xarxes MAN (Metropolitan Area Network) i WAN (Wide Area Network), per comunicacions via satèl·lit.

La plataforma que facilita el fabricant Digi per programar aquests mòduls s'anomena XCTU.

Aquesta plataforma permet decidir quin dels mòduls que es posen en el sistema dissenyat, vol que sigui el "Master" i els "Slave".

Per programar aquests dispositius se segueix el manual que facilita Digi en la seva web. En la Figura 5.12 es pot veure una imatge de la plataforma (interfície gràfica) XCTU amb els dos mòduls que es necessiten.

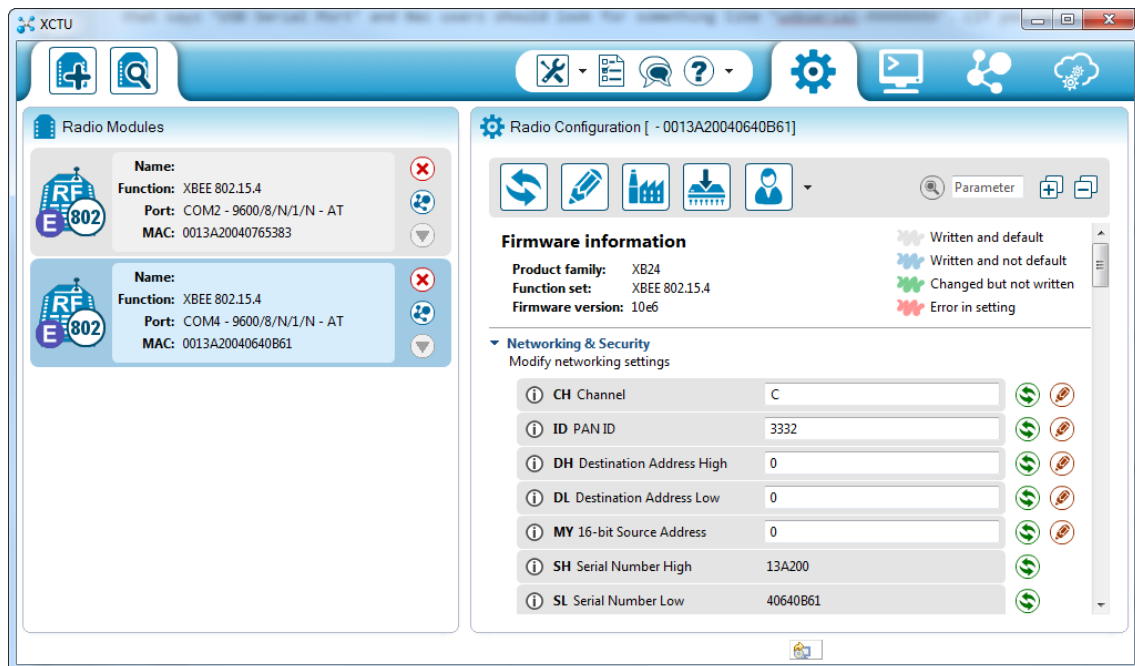


Figura 5.12. XCTU amb els dos Xbee.

Per establir la comunicació entre els dos mòduls es monta un amb la placa d'Arduino Uno sense el microcontrolador i s'afageix la shield que té Arduino model A000021 per a Xbee.

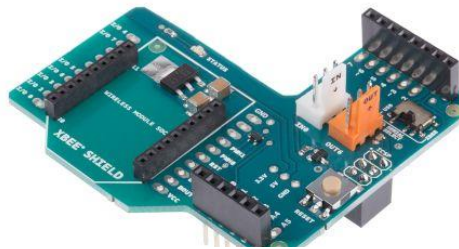


Figura 5.13. Mòdul A000021 per adaptar el Xbee a Arduino Uno.

Aquesta placa disposa d'un interruptor que s'haurà de posar en mode USB i que permetrà enviar comandes a través de la XCTU mitjançant el mòdul A000021 i el Xbee, a l'altre mòdul de situat al prototip II. La comunicació entre el Xbee i el dispositiu, i el Xbee i el ordinador, és per comunicació sèrie (UART). En la Figura 5.14 i 5.15 es mostra el funcionament de la comunicació amb Xbee S2C per radiofreqüència (RF).

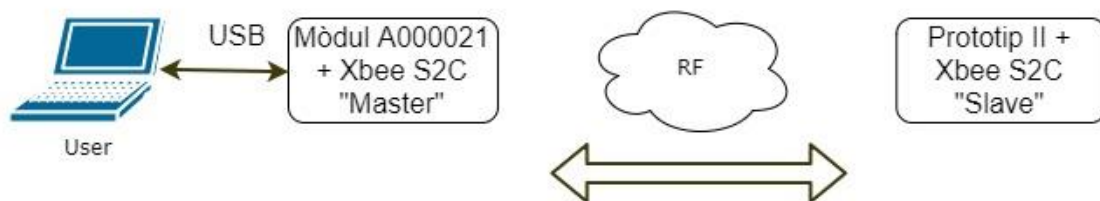


Figura 5.14. Esquema per al funcionament de la comunicació RF.

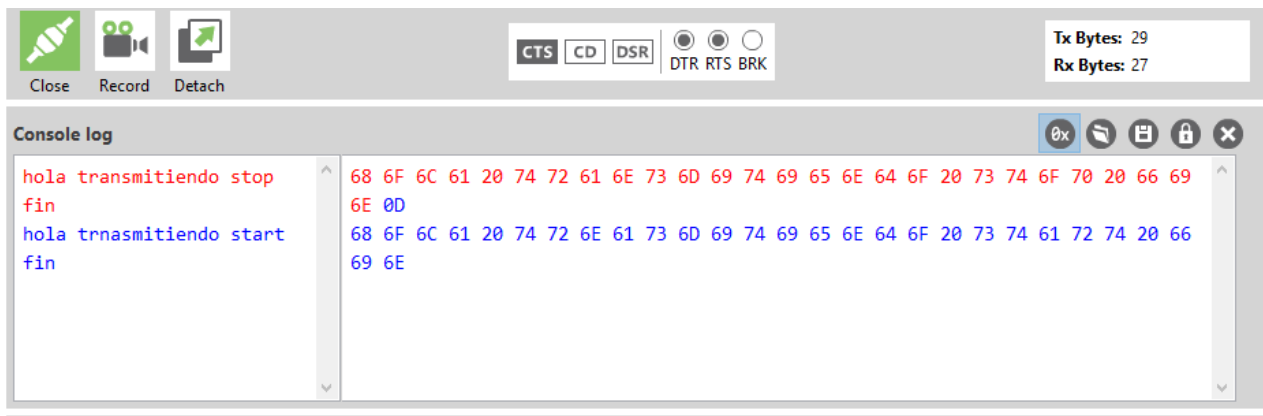


Figura 5.15. Comunicació a través del xat de XCTU. Missatge enviat (vermell) i missatge rebut (blau).

En el cas de la Figura 5.15 es mostra una imatge amb una consola d'un xat amb XCTU tipus eco. Els xats tipus eco simplement envien una comanda i reben la mateixa, ja que el dispositiu al que se l'hi envia té un pont en els canals de comunicació (RX i TX).

Per aquesta consola s'escriuran les variables per interactuar amb el microcontrolador ATmega328P amb el seu Xbee incorporat. Aquestes variables es troben a l'apartat: 5.2.4. Comandes registrades, i són: 'i', 'o', 'p', 'r', 't'.

Els pins del Xbee S2C es mostren a la Figura 5.16. Pel seu funcionament només es requerirà del pin 1,2,3 i 9.

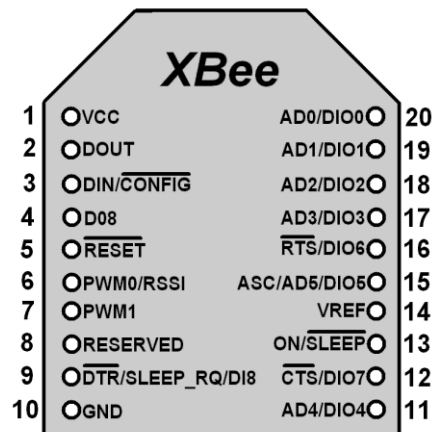


Figura 5.16. Pins Xbee S2C [57].

Pins Xbee S2C	Funció	Detall
1	Vcc	3,3 V
2	RX	Recepció de dades
3	TX	Transmissió de dades
10	GND	Massa

Taula 5.1. Pins requerits per al funcionament del Xbee.

S'ha decidit utilitzar aquest dispositiu ja que té una gran quantitat d'informació i llibreries per programar-lo, a més, disposa d'un microcontrolador propi programable (en aquest treball de fi de grau no es podia abastar). Finalment, el mòdul Xbee s'utilitza com un transmissor i receptor simple de radiofreqüència.

## 5.2.6. Esquema placa general

L'esquema del prototip II s'adjunta a l'Annex 1 i el material pel muntatge als PRESSUPOSTOS.

Tot i això, a continuació es detallen les parts més rellevants de l'esquemàtic i de la placa (PCB).

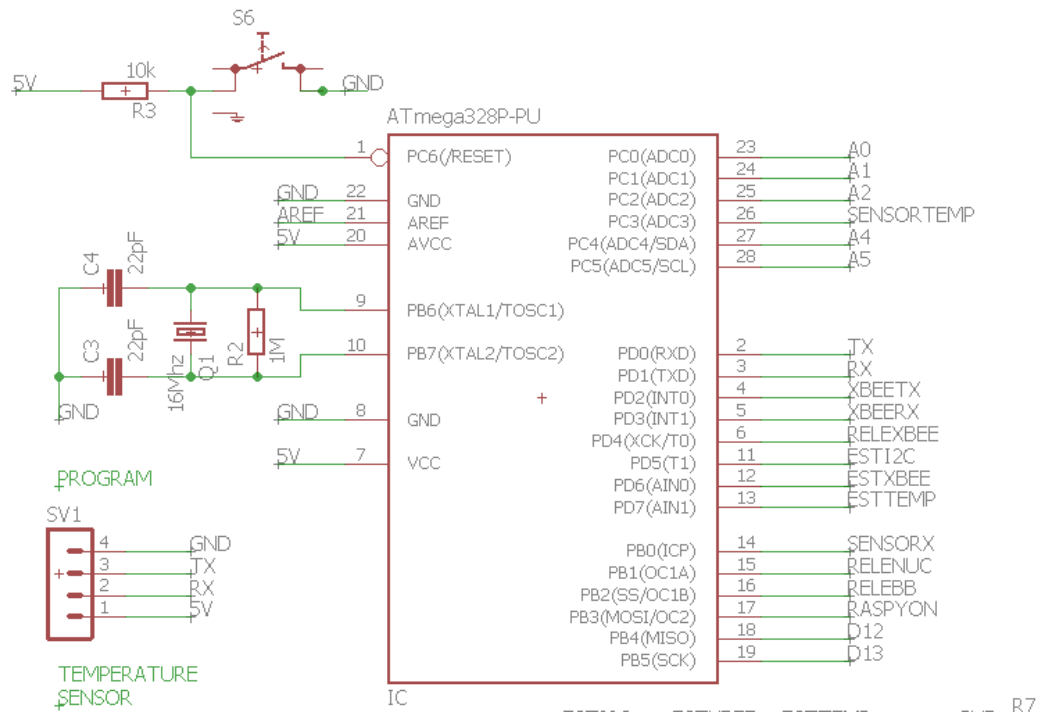


Figura 5.17. Conexió del ATmega328P-PU amb els pins referenciats.

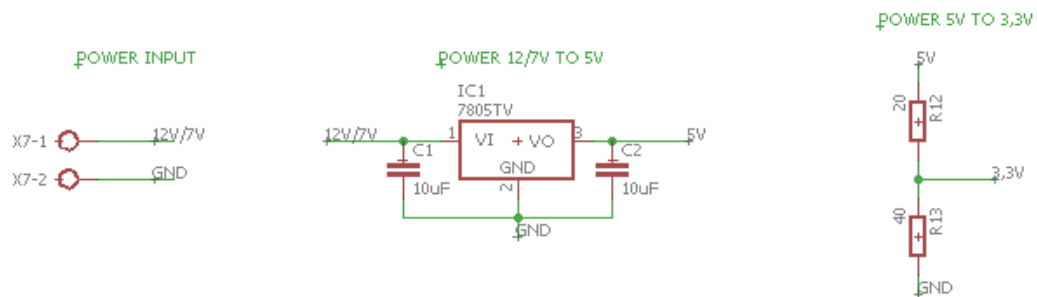


Figura 5.18. Power system.

El sistema compta amb un convertidor lineal LM7805 (DC7DC) que transforma un voltatge d'entrada d'entre 12V i 7V a 5V. Entrega una corrent màxima de 1,5 A, que ja és suficient per fer anar la placa ja que aquesta en els pitjors dels casos consumeix 600 mA.

Per l'alimentació del Xbee a 3,3 V es proposa un convertidor lineal però no es trobava a cap tenda de les consultades, inclús en físiques. Per tant, es fa un divisor de tensió (dreta) amb unes resistències de 20  $\Omega$  i 40  $\Omega$ .

Els càlculs que s'han dut a terme es detallen a continuació:

- 1) S'aplica la fórmula pel divisor de tensió on  $V_x$  és el voltatge de 3,3 V que volem obtenir i  $V_{cc}$  el voltatge amb el que alimentem al divisor de tensió, en aquest cas a 5 V.

$$V_x = V_{cc} \cdot \frac{R_{13}}{R_{13} + R_{12}} \quad (Eq. 1)$$

$$3,3 \text{ V} = 5 \cdot \frac{R_{13}}{R_{13} + R_{12}} \quad (Eq. 2)$$

$$R_{13} = 1,94 \cdot R_{12} \quad (Eq. 3)$$

$$R_{13} = 2 \cdot R_{12} \quad (Eq. 4)$$

- 2) Es dimensiona la  $R_{12}$  per una intensitat de 80 mA, ja que s'espera un consum d'uns 50 mA del Xbee.

$$R_{12} = \frac{1,7 \text{ V}}{0,080 \text{ A}} \quad (Eq. 5)$$

$$R_{12} = 21,25 \Omega$$

Si es busca al mercat la resistència més fàcil de trobar és de 20  $\Omega$ , per tant:

$$\mathbf{R_{12} = 20 \Omega}$$

- 3) Es troba  $R_{13}$ .

$$R_{13} = 2 \cdot R_{12} \quad (Eq. 6)$$

$$\mathbf{R_{13} = 40 \Omega}$$



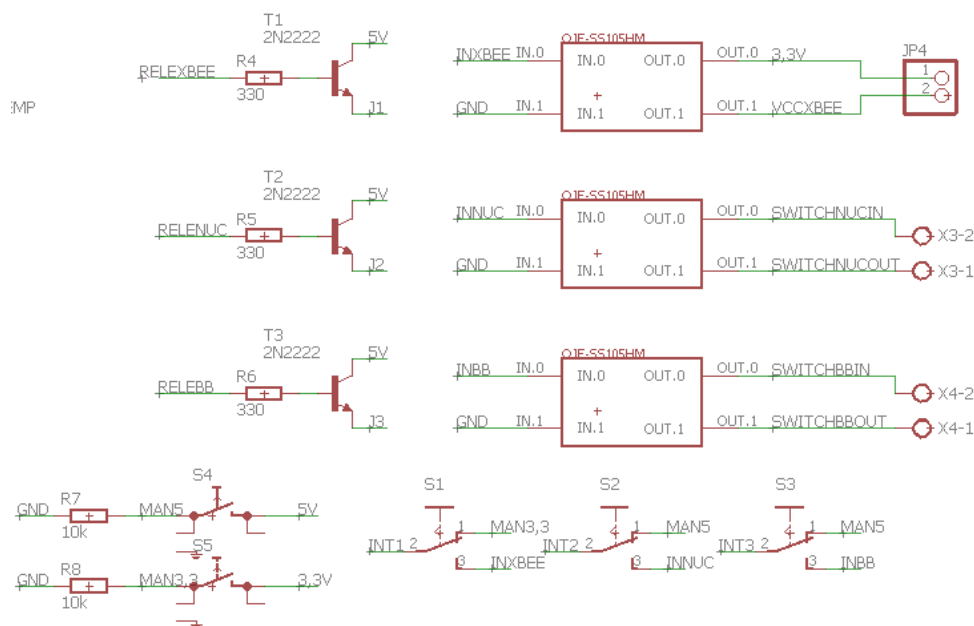
4) Es comprova que amb aquestes resistències dóna un valor de 3,3 V.

$$V_x = V_{cc} \cdot \frac{R_{13}}{R_{13} + R_{12}} \quad (Eq.7)$$

$$Vx = 5 \cdot \frac{40}{40 + 20} \quad (Eq.8)$$

$$V_x = 3,33 \text{ V}$$

Es dona per vàlid el resultat, tot i que el divisor de tensió és el mètode menys òptim per alimentar un dispositiu.



*Figura 5.19. Sistema de relés i interruptors (manual/automàtic).*

En la Figura 5.19 es pot veure els transistors amb els relés corresponents, ja que si es connecten directament els relés als transistors el microcontrolador no pot entregar suficient intensitat pel pin.

També a sota a l'esquerra s'hi troben els pulsadors que permeten emular el pulsador del dispositiu en cas de que els interruptors, situats a la dreta de l'esquema, es posin en mode manual.

Els tres interruptors tenen la funció de posar els relés en mode manual o automàtic. Aquests tres interruptors corresponen a cadascun dels tres relés (Xbee, NUC i BeagleBone).

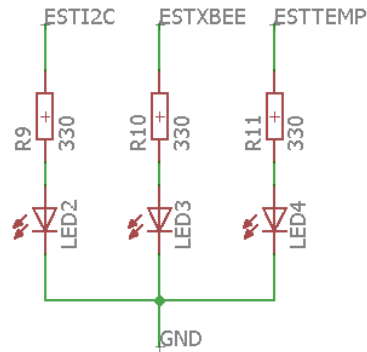


Figura 5.20. Leds indicadors d'estat. Groc, vermell i verd.

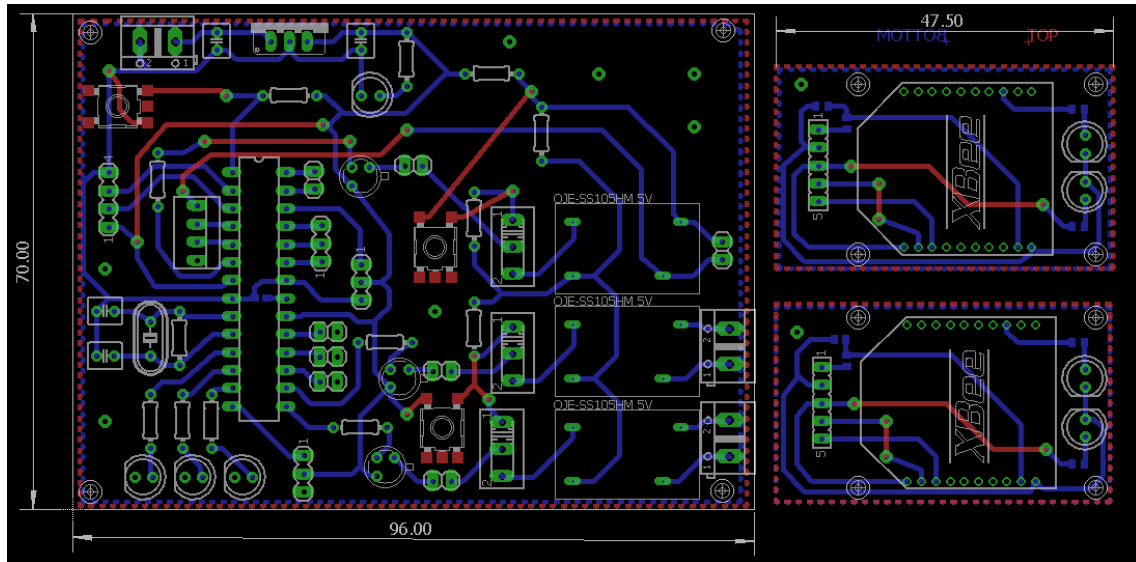


Figura 5.21. PCB amb la distribució dels components del prototip II.

A la Figura 5.21 s'hi troba la distribució dels components a la PCB. En concret, s'han realitzat tres dispositius en una sola placa PCB. A l'esquerra de la imatge s'hi troba el prototip II, i a la dreta dos adaptadors de pins del Xbee (es passa d'uns pins de 2 mm a uns de 2,54 mm) amb dos leds. Finalment no s'instal·laran ni les resistències ni els leds físicament ja que no són necessaris pel funcionament.

Les dimensions del prototip II són de 96 mm x 70 mm.

### 5.2.7. Problemes del prototip II

A continuació es detallen els problemes que s'han tingut en el desenvolupament del disseny del prototip II.

- Qualitat placa:

Al utilitzar-se una placa de fibra de vidre de doble cara, feta artesanalment en el taller de la universitat, els resultats no són els més bons. Tampoc ha ajudat no tenir el tamany de broques ideal.

- Xbee S2C:

Es va decidir triar aquest mòdul per la seva versalitat i preu, ara bé, la programació dels dispositius es va allargar fins a tres setmanes.

En concret, el model utilitzat és un Xbee S2C rev A (primera revisió), és a dir, els primers models que fabriquen.

Inicialment es va decidir comprar dos models. Un d'ells es programava sense problema i l'altre no es va poder, tot això mitjançant el XCTU (interfície) i el mòdul FT232RL o el CP2102.

Es va fer una segona compra amb tres mòduls més Xbee S2C, per intentar programar el mòdul que faltava per fer parella amb el primer.

Arribats els mòduls Xbee S2C (segona compra) es van testejar i es va aconseguir fer anar correctament dos dels tres. Conscient d'aquest fet, es va notificar a RS Components, web on s'havien comprat aquest dispositius i molt amablement em van facilitar dos nous.

En una de les compres s'havia adquirit el mòdul A000021 (Figura 5.13), així que es va intentar fer funcionar aquest mòdul mirant el seu datasheet. Doncs bé, després d'una intensa cerca, la sorpresa va ser que el datasheet d'aquest mòdul no està actualitzat, i conté les explicacions per al mòdul anterior. Per tant, no eren útils per al mòdul A000021, que disposa d'un interruptor amb dues posicions ("Xbee/USB").

Es van fer noves proves (enginyeria inversa) per descobrir el funcionament del mòdul A000021, i finalment es va arribar a la conclusió de que les posicions del interruptor indicaven si els pins 0 (RX) i 1 (TX) del Arduino Uno estan connectats directament o no amb el mòdul.

Així s'arriba a la conclusió de que si l'interruptor està en posició "USB", permet carregar un programa al ATmega328P-PU, en canvi si l'interruptor està en posició "Xbee", permet tenir el xbee activat.

En resum, si s'utilitza el mòdul A000021, primer es programa el ATmega328P-PU amb el pin situat en la posició "USB", i després per activar el Xbee, es mou l'interruptor en la posició "Xbee".

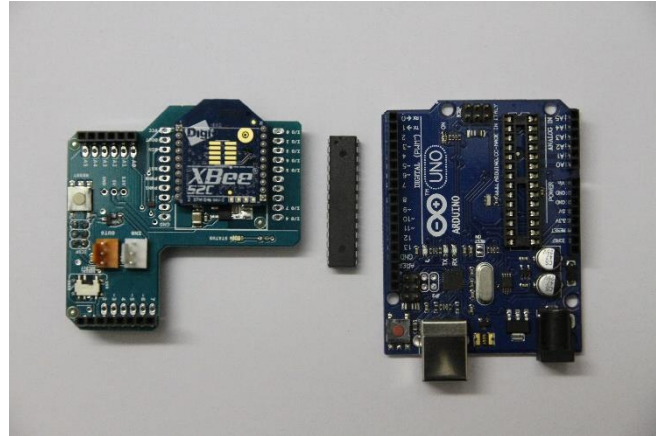
En aquest moment, es lliguen caps. Si es té en compte l'esquema que té la placa Arduino Uno (amb el ATmega328P-PU) i el mòdul A000021, s'arriba a la conclusió de que per poder programar el Xbee S2C a través del mòdul A000021 mitjançant el convertidor de USB a TTL propi del Arduino Uno, s'ha d'extreure el microcontrolador ATmega328P-PU de la placa (Arduino Uno) i posar l'interruptor en mode "USB".

Efectivament, un cop extret el microcontrolador ATmega328P i posat el mòdul A000021 amb el Xbee S2C, es detecta el Xbee, mitjançant el XCTU.

Ara bé, per acabar de donar més emoció a aquest apartat, resulta que tots els mòduls funcionen amb aquesta espècie de "Hackeig", cosa inexplicable. Arribant a la conclusió que és un tema de disseny del propi Xbee S2C.

Per finalitzar, es van enviar correus al fabricant (Digi) explicant la situació sense rebre cap tipus de resposta.

Per tant, amb el mòdul USB a TTL que va inclòs dins del Arduino Uno si que funcionen tots els mòduls Xbee S2C, en canvi, amb convertors USB a TTL específics no es detecten tots els Xbee S2C.



*Figura 5.22. Mòdul A000021 amb Xbee, ATmega328P-PU i Arduino Uno.*

- Pic de voltatge de la bobina dels relés:

Durant les proves de test final es va detectar que al canviar de mode manual a automàtic o viceversa, el microcontrolador es reiniciava. També succeïa quan l'interruptor es trobava amb mode manual i es premia el pulsador.

Mitjançant un oscil·loscopi es detecta que es produeix un pic de voltatge que fa reiniciar el microcontrolador ATmega328P-PU.



Figura 5.23. Pics voltatge prototip II. V: 1 V/DIV, H: 500 ns/DIV

Com es pot comprovar a la Figura 5.23, en el moment de prémer el polsador, provoca un voltatge màxim de 6,5 V i un mínim de 2,5 V. El funcionament del ATmega328P admet un voltatge entre 5,5 V i 2,8 V, per tant, és normal que es reinici.

Aquest problema és degut a que la bobina del relé provoca una pic de voltatge quan es canvia de posició l'interruptor, ja que no hi ha cap dispositiu per descarregar la corrent.

Per solucionar aquest problema es decideix implementar un diode zener paral·lel a cada la bobina dels relés. També es decideix posar un condensador de 47 uF per evitar els rebots al prémer el polsador i d'altres petites interferències. Per últim, s'introdueix un condensador de desacoplament al microcontrolador més gran, en concret un de 10 uF electrolític i un de 100 nF ceràmic (Figura 5.24).

A la Figura 5.27 es poden veure amb vermell aquests elements. Els condensadors, els pulsadors i els relés, així com el diode zener, només estan aplicats en un dels casos. És a dir, que només estan aplicats en sistema hardware de la NUC (que inclou un relé, un interruptor i un pulsador), així durant la presentació es poden mostrar els errors comesos i les solucions adoptades.

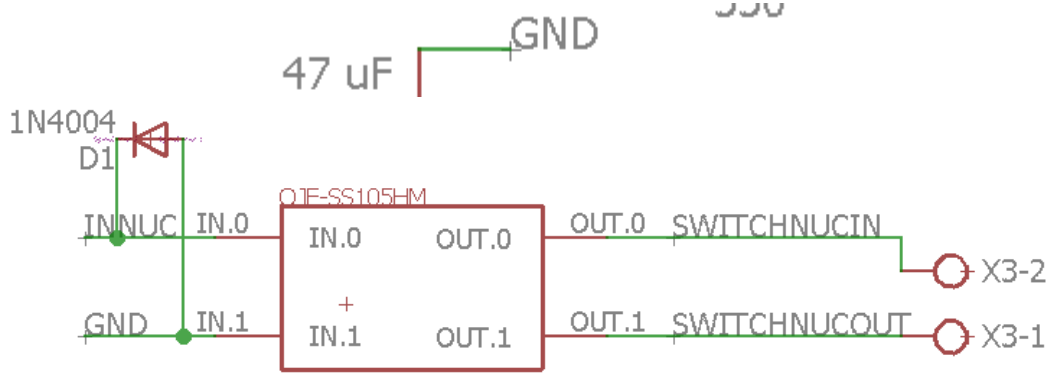


Figura 5.25. Solució pels pics de voltatge.

- Col·locació dels interruptors:

Es va posar malament els pins dels interruptors. A la Figura 5.26 (dreta), es pot veure com queden realment.

Per solucionar aquest problema s'han soldat uns cables multifilars en els pins de la PCB.



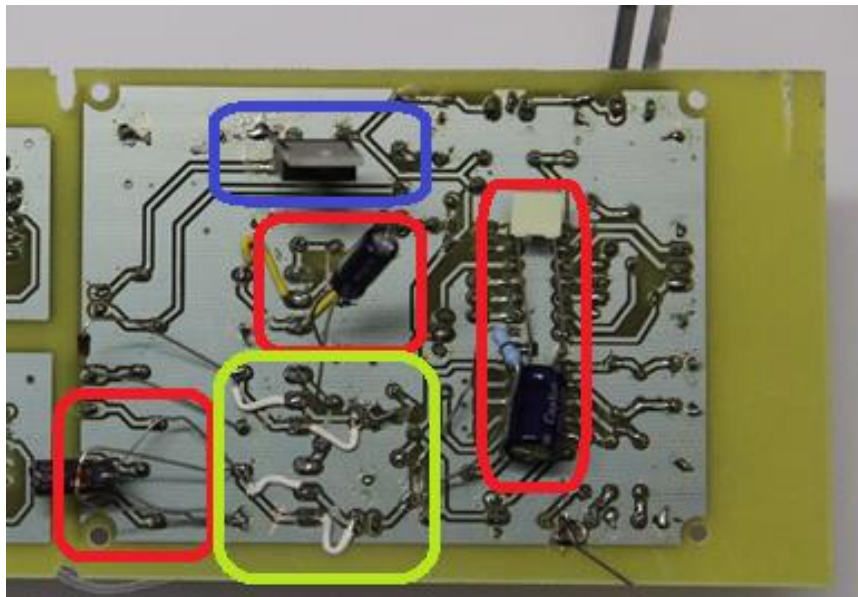
Figura 5.26. Interruptor erroni (esquerra) i interruptor correcte (dreta).

A la Figura 5.27 en verd es poden veure els interruptors modificats.

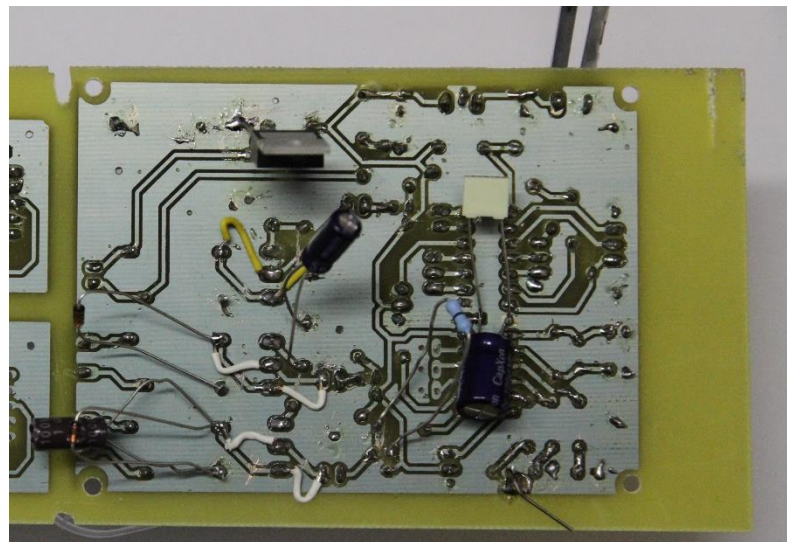
- Conversor linial a 3,3 V LM7803

Finalment es troba, ja que durant la realització d'aquest apartat no es disposava del material, un conversor linial model LM7803 de 3,3 V i 1 A de sortida i se solda a la placa evitant posar el divisor de tensions calculat anteriorment.

A la Figura 5.27 en blau es pot veure on està situat.



*Figura 5.27. Situació de nous components al prototip II.*



*Figura 5.28. "Bottom" de la PCB.*



### 5.3. Resultat final del prototip I i prototip II

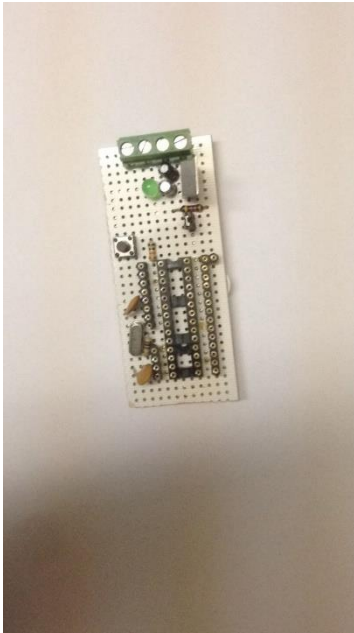


Figura 5.29. Prototip I.

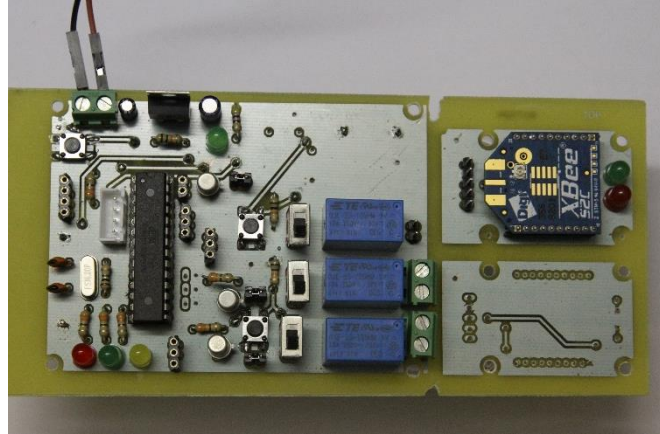


Figura 5.30. Prototip II amb el Xbee.

La placa del prototip II està llesta per ser tallada on es mostren les dos esquerdes.

### 5.4. Conclusions

S'han treballat els camps del software i hardware, aportant una solució a un problema com és l'encesa dels diferents dispositius de la nova arquitectura del Guanay 3.0.

També s'ha arribat a solucionar tots els problemes que han aparegut durant el disseny i tests del prototip I i II.

Per últim, es voldria aclarar que aquest disseny no és el definitiu del projecte real del Guanay 3.0, sinó que estudia una possible solució. Caldria fer la PCB a fàbrica, cosa que no s'ha pogut fer en aquest treball de final de grau degut a les dificultats que s'han trobat per part de la universitat. D'aquesta manera es poden optimitzar els components i espai, substituint components amb un encapsulat d'inserció per components SMD, així com fer ús d'optoacopladors en comptes de relés, entre d'altres canvis.

El motiu pel qual s'han utilitzat els components d'aquest disseny és degut a la falta de material al laboratori, així doncs, s'han utilitzat uns components el més econòmics possibles.



Si més no, aquest disseny d'encesa aporta una idea de com hauria de ser el sistema real per a l'encesa dels diferents dispositius en el vehicle Guanay 3.0 .

## CONCLUSIONS FINALS

En el treball de fi de grau d'Enginyeria Electrònica Industrial i Automàtica, s'han complert tots els objectius que s'exposaven al principi del treball sobre l'estudi i l'adaptació d'una unitat de control.

S'ha treballat en diferents camps de l'electrònica, des de la recerca fins al disseny i implementació d'un model electrònic, en aquest cas, tant en l'àmbit del software com del hardware.

També s'ha après a utilitzar eines de disseny de PCB, simuladors de circuits, plataformes de programació i altres elements, que han requerit de esforç, temps i també de frustracions en alguns moments, però que finalment, amb esforç, temps i paciència s'ha aconseguit solventar els problemes de manera satisfactòria.

El treball s'ha centrat finalment en els següents aspectes:

- Recerca i tractament veraç de la informació.
- Utilització i recerca d'eines Open Source.
- Adaptació d'una unitat de control d'emergència.
- Aportar solucions a problemes presents o futurs de les unitats de control.
- Introducció a un nou projecte, el Guanay 3.0.
- Dissenyar des de zero un prototip electrònic.
- Verificar el disseny del prototip.

A l'inici del projecte, es plantejava com una adaptació de la Intel NUC5i7RYH al Guanay II, però, degut a la falta de recanvis del propi vehicle, les missions constants del vehicle i la distància on estava ubicat, es va decidir canviar el rumb del projecte, donant-li un enfocament diferent.

En aquest punt es va decidir fer una investigació a fons dels diferents sistemes operatius, software i hardware, per determinar si la opció més correcte passava per fer l'adaptació o no.

A continuació, es va fer l'estudi sobre l'adaptació de la Intel NUC al Guanay II. Aquesta adaptació s'anomena Guanay 2.1 i, es planteja finalment com una unitat de control d'emergència.

En aquest punt, es va decidir plantejar una renovació completa del sistema de control de l'electrònica del Guanay II, passant a anomenar-se Guanay 3.0. Bàsicament es focalitza en el disseny d'una nova arquitectura amb un sistema de seguretat, per evitar que el vehicle es quedi sense cap mena de control en cas de fallada a la unitat de control principal.

Aquest projecte (Guanay 3.0) plantejava una sèrie de nous requisits, per tant, es decideix fer una investigació, desenvolupament i innovació d'un dels nous requisits, en aquest cas, un sistema d'encesa per als diferents dispositius del vehicle.

Amb tot això, es completa un cicle que permet treballar en diferents camps de l'electrònica, des de la cerca d'informació, fins al disseny del prototip i implementació d'aquest.

Cal dir que els nous dispositius i estructures electròniques del Guanay 2.1 i Guanay 3.0 permeten millorar les funcionalitats actuals del Guanay II.

Finalment, el ventall de conceptes d'electrònica treballats i desenvolupats en aquest projecte han permès adquirir una diversitat de coneixements molt útils per poder triar l'àmbit més adient en una propera carrera professional.

## PRESSUPOSTOS

Aquest apartat inclou els diferents pressupostos per:

- 1. Guanay 2.1
- 2. Guanay 3.0
- 3. Prototip I
- 4. Prototip II

### 1. Guanay 2.1

Per fer una valoració de costos s'ha realitzat una taula on es resumeix amb detall els costos del projecte, des de l'estudi que s'ha fet en aquest capítol i els dos primers, fins a l'adaptació del "codi" del Guanay II i adaptar-lo a la NUC5i7RYH. També s'ha tingut en compte, tant el personal com el material.

El sensors que s'han d'adaptar compten com a cost 0, ja que se suposa que seran els que hi ha actualment al Guanay II, no obstant això, es recomana guardar una part de pressupost per noves compres de sensors. En la següent taula es datallen els costos.

Material	Preu (€)
NUC5i7RYH	450
Disc dur HDD 1TB	100
RAM 4 GB	90
Hubs RS232	50
Hubs USB	20
Llicència Labview Professional	8.000
Llicència Windows 10 Pro	259
Fase I *	7.200
Fase II **	10.800
Altres	2.000

<b><u>TOTAL</u></b>	<b><u>28.969</u></b>
---------------------	----------------------

Taula P.1. Costos Guanay 2.1.

\*La **Fase I** és la fase amb el personal i les hores que s'han necessitat per fer l'estudi i fase pràctica d'aquest capítol i els dos anteriors. Es desglossa de la següent manera:

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 2

$$\text{Pressupost Fase I} = \text{Enginyer} \frac{\text{€}}{h} \cdot \frac{\text{Hores}}{\text{dia}} \cdot 30 \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase I} = 15 \frac{\text{€}}{h} \cdot 8 \frac{h}{\text{dia}} \cdot 2 \text{ mesos} \cdot 30 \text{ dies} = 7.200 \text{ €}$$

\*\*La **Fase II** és la fase amb el personal que es necessitaria per implementar i adaptar tot el codi del Guanay II (PC104+) al Guanay 2.1 (Intel NUC5i7RYH). En aquest apartat, es té en compte adaptacions de nous components com GPS, sensors i d'altres.

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 3

$$\text{Pressupost Fase II} = \text{Enginyer} \frac{\text{€}}{h} \cdot \frac{\text{Hores}}{\text{dia}} \cdot 30 \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase II} = 15 \frac{\text{€}}{h} \cdot 8 \frac{h}{\text{dia}} \cdot 3 \text{ mesos} \cdot 30 \text{ dies} = 10.800 \text{ €}$$

Cal remarcar que els costos de personal són orientatius i en bruts. Si es volgués comptar amb estudiantat o professorat propi, el cost es redueix al material i llicències.

## 2. Guanay 3.0

Per fer una valoració es fa una taula on es resumeix amb detall els costos de tot el projecte, des de l'estudi que s'ha fet en aquest capítol fins a l'adaptació del "codi" del Guanay II i adaptar-lo a la NUC5i7RYH, desenvolupar el sistema d'encesa (ATmega+Xbee), el sistema de seguretat (Raspberry Pi 3+), l'adquisició de dades i establir els protocols de comunicació. També s'ha tingut en compte tant el personal com el material.

En aquest cas no es té en compte el cost complet dels sensors o altres elements com el pistó i motors. Per això, es recomana deixar un pressupost aproximadament d'uns 5.000 € extra, inclosos dins el pressupost.

Material	Preu (€)
NUC5i7RYH	450
Disc dur HDD 1TB	100
RAM 4 GB	90
Hubs RS232	100
Hubs USB	20
Cablejat	150
Subjecció unitat de control	200
Raspberry Pi 3	40
Beagle Bone Black	50
Sistema encesa (inclou prototip I i prototip II)	100
Fase I *	7.200
Fase II **	172.800
Fase III***	3.600
Altres	5.000

<b><u>TOTAL</u></b>	<b><u>189.900</u></b>
---------------------	-----------------------

Taula P.2. Pressupost final Guanay 3.0.

\*La **Fase I** és la fase on el personal i hores que s'ha necessitat per fer l'estudi d'aquest capítol i dels dos primers, i es desglossa de la següent manera:

Enginyer= 15 €/h

Personal= 1

Hores: 8 h/dia

Mesos: 2

$$\text{Pressupost Fase I} = \text{Enginyer} \frac{\text{€}}{h} \cdot \frac{\text{Hores}}{\text{dia}} \cdot 30 \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase I} = 15 \frac{\text{€}}{h} \cdot 8 \frac{h}{\text{dia}} \cdot 2 \text{ mesos} \cdot 30 \text{ dies} = 7.200 \text{ €}$$

\*\*La **Fase II** és la fase on el personal que es necessitaria per implementar i adaptar tot el codi del Guanay II (PC104+), al Guanay 3.0. (Intel NUC5i7RYH), desenvolupar tot el codi pel sistema de seguretat (Raspberry Pi3+), l'adquisició de dades (Beagle Bone) i el sistema d'encesa (ATmega+Xbee). En aquest apartat també es té en compte adaptacions de nous components com GPS, sensors i d'altres.

Enginyer= 15 €/h

Personal= 3 Enginyers

Hores: 8 h/dia

Mesos: 16

$$\text{Pressupost Fase II} = \text{Enginyer} \frac{\text{€}}{h} \cdot \frac{\text{Hores}}{\text{dia}} \cdot 30 \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase II} = 3 * 15 \frac{\text{€}}{h} \cdot 8 \frac{h}{\text{dia}} \cdot 16 \text{ mesos} \cdot 30 \text{ dies} = 172.800 \text{ €}$$

\*\*\*La **Fase III** és la fase on el personal s'encarregarà d'adaptar definitivament tot el material dins del Guanay 3.0.

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 1

$$\textit{Pressupost Fase III} = \textit{Enginyer} \frac{\text{€}}{h} \cdot \frac{\text{Hores}}{\text{dia}} \cdot 30 \frac{\text{dies}}{\text{mes}}$$

$$\textit{Pressupost Fase III} = 15 \frac{\text{€}}{h} \cdot 8 \frac{h}{\text{dia}} \cdot 1 \text{ mes} \cdot 30 \text{ dies} = \mathbf{3.600 \text{ €}}$$

Cal remarcar que els costos de personal són orientatius i bruts. Si es volgués comptar amb estudiant o professorat propi, el cost es redueix al material.



### 3. Prototip I

En aquest apartat es tindrà en compte el material i el personal necessari per al muntatge del prototip I.

Material	Unitats	Preu unitat (€)	Preu final (€)
ATmega328P-PU	1	3	3
Sòcol 28 pins	1	0,50	0,50
Tira de 40 pins	1	2	2
Cristall 16 MHz	1	0,50	0,50
Condensador 22pF	2	0,15	0,30
Condensador 10 uF	2	0,30	0,60
LM7805C	1	1,15	1,15
Resistència 10.000 $\Omega$	1	0,05	0,05
Resistència 220 $\Omega$	1	0,05	0,05
Resistència 330 $\Omega$	1	0,05	0,05
Led verd	2	0,10	0,20
Connector 5mm 2 pos.	2	0,80	1,60
Placa baquelita	1	2,50	2,50
Jumper	1	0,10	0,10
FT232RL	1	5	5
Cablejat	1	5	5
Fase I *	-	-	1.800
Fase II **	-	-	1.800

<b><u>TOTAL AMB PERSONAL</u></b>	<b><u>3622,6</u></b>
<b><u>TOTAL SENSE PERSONAL</u></b>	<b><u>22,6</u></b>

Taula P.3. Costos prototip I.

\*La **Fase I** és la fase amb el personal i les hores que s'han necessitat per fer l'estudi i fase pràctica d'aquest prototip I. Es desglossa de la següent manera:

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 1/2

$$\text{Pressupost Fase I} = \text{Enginyer} \frac{\text{€}}{\text{h}} \cdot \frac{\text{Hores}}{\text{dia}} \cdot \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase I} = 15 \frac{\text{€}}{\text{h}} \cdot 8 \frac{\text{h}}{\text{dia}} \cdot 15 \text{ dies} = \mathbf{1.800 \text{ €}}$$

\*\*La **Fase II** és la fase que requereix de muntatge i test de la placa.

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 1/2

$$\text{Pressupost Fase II} = \text{Enginyer} \frac{\text{€}}{\text{h}} \cdot \frac{\text{Hores}}{\text{dia}} \cdot \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase II} = 15 \frac{\text{€}}{\text{h}} \cdot 8 \frac{\text{h}}{\text{dia}} \cdot 15 \text{ dies} = \mathbf{1.800 \text{ €}}$$

Cal remarcar que els costos de personal són orientatius i en brut. Si es volgués comptar amb estudiantat o professorat propi, el cost es redueix al material.

#### 4. Prototip II

En aquest apartat es tindrà en compte el material i el personal necessari per al muntatge del prototip II.

Material	Unitats	Preu unitat (€)	Preu final (€)
Components I*	-	22,6	22,6
PCB doble cara	1	0,50	8
Tira de 40 pins	1	2	2
Xbee S2C	2	18	36
Interruptors	3	0,50	1,50
Polsadors	2	0,75	1,50
Relé 5V	3	0,50	1,50
Condensador 47 uF	3	0,20	0,60
Diode 12 V	3	0,20	0,60
Resistència 10.000 $\Omega$	5	0,05	0,25
Resistència 1 M $\Omega$	1	0,05	0,05
Resistència 330 $\Omega$	10	0,05	0,50
Resistència 20 $\Omega$	1	0,05	0,05
Resistència 40 $\Omega$	1	0,05	0,05
Led verd	1	0,10	0,10
Led groc	1	0,10	0,10
Led vermell	1	0,10	0,10

Jumper	4	0,10	0,40
Cablejat	1	1,5	1,5
Fase I **	-	-	1.800
Fase II ***	-	-	1.800
<b><u>TOTAL AMB PERSONAL</u></b>			<b><u>3677,5</u></b>
<b><u>TOTAL SENSE PERSONAL</u></b>			<b><u>77,45</u></b>

Taula P.4. Material pel prototip II.

\* Els **Components I** són els mateixos components que s'utilitzen per al prototip I. Només es tindrà en compte el cost de material.

\*\*La **Fase I** és la fase amb el personal i les hores que s'han necessitat per fer l'estudi i fase de disseny d'aquest prototip I. Es desglossa de la següent manera:

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 1/2

$$\text{Pressupost Fase I} = \text{Enginyer} \frac{\text{€}}{\text{h}} \cdot \frac{\text{Hores}}{\text{dia}} \cdot \frac{\text{dies}}{\text{mes}}$$

$$\text{Pressupost Fase I} = 15 \frac{\text{€}}{\text{h}} \cdot 8 \frac{\text{h}}{\text{dia}} \cdot 15 \text{ dies} = 1.800 \text{ €}$$

\*\*\*La **Fase II** és la fase que requereix de muntatge i test de la placa.

Enginyer= 15 €/h

Personal= 1 Enginyer

Hores: 8 h/dia

Mesos: 1/2

$$\textit{Pressupost Fase II} = \textit{Enginyer} \frac{\text{€}}{h} \cdot \frac{\text{Hores}}{\text{dia}} \cdot \frac{\text{dies}}{\text{mes}}$$

$$\textit{Pressupost Fase II} = 15 \frac{\text{€}}{h} \cdot 8 \frac{h}{\text{dia}} \cdot 15 \text{ dies} = \mathbf{1.800 \text{ €}}$$

Cal remarcar que els costos de personal són orientatius i en brut. Si es volgués comptar amb estudiantat o professorat propi, el cost es redueix al material.

## BIBLIOGRAFIA

### CAPÍTOL 1

[1] González Agudelo, J (2015). *Contribution to the model and navigation control of an autonomous underwater vehicle*. Universitat Politècnica de Catalunya. Departament d'Enginyeria Electrònica. Disponible a: <https://upcommons.upc.edu/bitstream/handle/2117/95737/TJGA1de1.pdf> [últim accés desembre 2017].

[2] Masmitjà i Rusinyol, I (2013). *Tesi final de màster: aportacions realitzades al vehicle Guanay II AUV* [Internet]. Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona. Universitat Politècnica de Catalunya. Disponible a: [https://upcommons.upc.edu/bitstream/handle/2099.1/19145/Aportacions\\_Realitzades\\_al\\_Vehicl\\_e\\_GuanayII\\_AUV\\_\(Ivan\\_Masmitja\)R3.pdf](https://upcommons.upc.edu/bitstream/handle/2099.1/19145/Aportacions_Realitzades_al_Vehicl_e_GuanayII_AUV_(Ivan_Masmitja)R3.pdf) [últim accés desembre 2017].

[3] Navaldrone.com. [Internet]. Disponible a: <http://www.navaldrone.com/Remus.html> [últim accés octubre 2017].

[4] Nauticexpo.es. [Internet]. Disponible a: <http://www.nauticexpo.es/prod/saab-seaeye/product-25353-182635.html> [últim accés octubre 2017].

[5] Offshorewind.biz. AUV [Internet]. Disponible a: <https://www.offshorewind.biz/tag/auvs/> [últim accés octubre 2017].

[6] Auvac.org. AUVAC: *Strengthening the AUV community* [Internet]. Disponible a: <http://auvac.org/> [últim accés octubre 2017].

[7] Rudnick D, Davis R, Eriksen C, Fratantoni D, Perry MJ (2004). *Underwater gliders for Ocean Research* [Internet]. Marine Technology Society Journal. Vol 38 (1). Disponible a: [http://pordlabs.ucsd.edu/rdavis/publications/MTS\\_Glider.pdf](http://pordlabs.ucsd.edu/rdavis/publications/MTS_Glider.pdf) [últim accés octubre 2017].

[8] Claustre H, Beguery L, Pla Patrice. *SeaExplorer Glider Breaks Two World Records* [Internet]. Sea Technology Magazine. Disponible a: <https://www.sea-technology.com/features/2014/0314/3.php> [últim accés octubre 2017].

[9] Tarabuta O, Ichimoei G, Clinci CP, Gherghina A (2016). *Biomimetic control of autonomous underwater vehicles: a modelling algorithm for computing the current sway* [Internet]. "Mircea cel Batran" Naval Academy Scientific Bulletin. Vol 19 (2): 103-108. Disponible a:

[https://www.anmb.ro/buletinstiintific/buletine/2016\\_Issue2/NMS/103-108.pdf](https://www.anmb.ro/buletinstiintific/buletine/2016_Issue2/NMS/103-108.pdf) [últim accés octubre 2017].

[10] Sbir.gov (2011). *BIOSwimmer, a Biomimetic, Autonomous, Unmanned Underwater Vehicle (UUV)* [Internet]. SBIR-STTR. Disponible a: <https://www.sbir.gov/node/735801> [últim accés octubre 2017].

[11] Lynceans.org (2016). *Wave Glider Autonomous Vehicle Harvests Wave and Solar Power to Deliver Unique Operational Capabilities at Sea*. Lynceans group. Disponible a:

<http://www.lynceans.org/all-posts/wave-glider-autonomous-vehicle-harvests-wave-and-solar-power-to-deliver-unique-operational-capabilities-at-sea/> [últim accés octubre 2017].

[11] Otero, M (2014). *Submarino S-41 ARA Santa Cruz*. [Internet]. Disponible a: <http://www.panoramio.com/photo/108235659> [últim accés desembre 2017].

## CAPÍTOL 2

[12] Informatica-hoy.com.ar. *Que es el sistema operativo?* [Internet]. Disponible a: <https://www.informatica-hoy.com.ar/aprender-informatica/Que-es-el-sistema-operativo.php> [últim accés novembre 2017].

[13] Microsoft.com. (2017). *Windows / Sitio oficial del sistema operativo Microsoft Windows 10 Home, S y Pro, portátiles, tabletas y mucho más* [Internet]. Disponible a: <https://www.microsoft.com/es-es/windows/> [últim accés desembre 2017].

[14] Apple.com.es. [Internet]. Disponible a: <https://www.apple.com/es/> [últim accés novembre 2017].

[15] Elinux.org (2017). *Beagleboard: BeagleBoneBlack* [Internet]. Disponible a: <https://elinux.org/Beagleboard:BeagleBoneBlack> [últim accés novembre 2017].

[16] Freertos.org. *The FreeRTOS Kernel: Market Leading, De-facto Standard and Cross Platform RTOS Kernel* [Internet]. Disponible a: <https://www.freertos.org/> [últim accés novembre 2017].

[17] Xilinx.com. *Xilinx All programmable*. [Internet]. Disponible a: <https://www.xilinx.com/> [últim accés desembre 2017].

[18] Es.rs-online.com. *RS Components / Componentes Electrónicos y Eléctricos* [Internet]. Disponible a: <https://es.rs-online.com/web/> [últim accés desembre 2017].



- [19] Microchip.com. *ATmega328p in Production* [Internet]. Disponible a: <http://www.microchip.com/wwwproducts/en/ATmega328p> [últim accés desembre 2017].
- [20] Ti.com. *RMS320F28335 (Active): Delfino Microcontroller* [Internet]. Disponible a: <http://www.ti.com/product/TMS320F28335> [últim accés novembre 2017].
- [21] Ti.com. *Stellaris LM4F120 Series MCU* [Internet]. Disponible a: <http://www.ti.com/en/graphics/mcu/stellaris-arm-cortex-m4-LM4F120.jpg> [últim accés novembre 2017].
- [22] Ihs (2015). *Top 10 Ranked MCU Suppliers Assuming Microchip's Acquisition of Atmel*. IHS. Disponible a: [https://m.eet.com/images/eetimes/2016/03/1329187/IHS\\_pie03\\_Top\\_10\\_MCU\\_suppliers.png](https://m.eet.com/images/eetimes/2016/03/1329187/IHS_pie03_Top_10_MCU_suppliers.png) [últim accés novembre 2017].
- [23] Pinterest.es. *Von Neumann Architecture: The Reference Model for Computer* [Internet]. Pinterest. Disponible a: <https://www.pinterest.es/pin/24136547976279563/> [últim accés novembre 2017].
- [24] Raspberrypi.org [Internet]. Disponible a: <https://www.raspberrypi.org/> [últim accés desembre 2017].
- [25] Debian.org. (2017). *Debian: the universal operating system* [Internet]. Disponible a: <https://www.debian.org/> [últim accés novembre 2017].
- [26] Element14.com (2015). *Raspberry Pi 3 Model B Gpio 40 Pin Block Pinout* [Internet]. Disponible a: <https://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-3-model-b-gpio-40-pin-block-pinout> [últim accés desembre 2017].
- [27] BeagleBoard.org (2017). *BeagleBone: open-hardware expandable computer* [Internet]. Disponible a: <http://beagleboard.org/support/bone101> [últim accés novembre 2017].
- [28] Intel.es. *Kit Para Intel: NUC NUC5i7RYH* [Internet]. Disponible a: <https://www.intel.es/content/www/es/es/products/boards-kits/nuc/kits/nuc5i7ryh.html> [últim accés novembre 2017].
- [29] Asus.com (2015). *VivoMini UN65: ultra compact PC for exceptional desktop performance* [Internet]. Disponible a: <https://www.asus.com/us/Mini-PCs/VivoMini-UN65/> [últim accés novembre 2017].

- [30] Amazon.es (2017). *Asus UN65 M025 LGA 1356 (Socket B2) 2.5GHz i7-6500U 0.79l tamaño PC Azul-Barebón (LGA 1356 (Socket B2), 6ª generación de procesadores Intel Core i7, 2.5GHz, i7-6500U, 14nm, 3.1GHz)* [Internet]. Disponible a: [https://www.amazon.es/UN65-M025M-Socket-2-5GHz-i7-6500U-tama%C3%B1o/dp/B01C3NR4U0/ref=sr\\_1\\_3?ie=UTF8&qid=1514478503&sr=8-3&keywords=mini%2Bpc%2Bintel%2Bi7&th=1](https://www.amazon.es/UN65-M025M-Socket-2-5GHz-i7-6500U-tama%C3%B1o/dp/B01C3NR4U0/ref=sr_1_3?ie=UTF8&qid=1514478503&sr=8-3&keywords=mini%2Bpc%2Bintel%2Bi7&th=1) [últim accés novembre 2017].
- [31] Microchip.com. *Microchip Technology* [Internet]. Disponible a: <http://www.microchip.com> [últim accés novembre 2017].
- [32] Nxp.com. *NXP Semiconductors* [Internet]. Disponible a: <https://www.nxp.com/> [últim accés novembre 2017].
- [33] Kozub, R (2013). *FreeMASTER Usage. Serial driver implementation* [Internet]. Roznov: Freescale Semiconductor. Disponible a: <https://www.nxp.com/docs/en/application-note/AN4752.pdf> [últim accés novembre 2017].
- [34] Freescale Semiconductor Inc (2012). *MC9S08PT60 Series* [Internet]. Freescale Semiconductor. Disponible a: <http://docs-europe.electrocomponents.com/webdocs/12b6/0900766b812b6910.pdf> [últim accés novembre 2017].
- [35] Montañez, E (2004). *Using the HCS08 Family On-Chip In-Circuit Emulator (ICE)* [Internet]. Austin: Freescale Semiconductor. Disponible a: <https://www.nxp.com/docs/en/application-note/AN2596.pdf> [últim accés novembre 2017].
- [36] Nordic Semiconductor (2015). *nRF8001. Single-chip Bluetooth® low energy solution* [Internet]. Disponible a: <http://docs-europe.electrocomponents.com/webdocs/1429/0900766b81429d60.pdf> [últim accés novembre 2017].
- [37] Electroiq.com (2015). *Microcontroller Market Growth tied to IoT applications, IHS* [Internet]. Solid State Technology. Disponible a: <http://electroiq.com/blog/2015/09/microcontroller-market-growth-tied-to-iot-applications-ihs/> [últim accés novembre 2017].
- [38] Kranz M, Petisco JM (2017). *Internet of Things* [Internet]. Amazon.es. Disponible a: [https://www.amazon.es/Internet-Things-colecci%C3%B3n-acci%C3%B3n-empresarial/dp/8416894884/ref=sr\\_1\\_1?s=books&ie=UTF8&qid=1515146789&sr=1-1&keywords=internet+of+things](https://www.amazon.es/Internet-Things-colecci%C3%B3n-acci%C3%B3n-empresarial/dp/8416894884/ref=sr_1_1?s=books&ie=UTF8&qid=1515146789&sr=1-1&keywords=internet+of+things) [últim accés novembre 2017].

### CAPÍTOL 3

[39] Pc104.org. *PC Consortium* [Internet]. Disponible a: <https://pc104.org/> [últim accés desembre 2017].

[40] Rtd.com. *PC104 Modules, PC104 Systems, SBCs, Enclosures: RTD* [Internet]. Disponible a: <https://www.rtd.com> [últim accés octubre 2017].

[41] Pc104.org (2008). *PC104 Specification. Version 2.6* [Internet]. Pc104. Disponible a: [https://pc104.org/wp-content/uploads/2015/02/PC104\\_Spec\\_v2\\_6.pdf](https://pc104.org/wp-content/uploads/2015/02/PC104_Spec_v2_6.pdf) [últim accés desembre 2017].

[42] Yee A (2016). *Intel NUC5i7RYH review: The poster child of mini-PCs still leads the pack.* [Internet]. PCWorld. Disponible a: <https://www.pcworld.com/article/3039427/hardware/intel-nuc5i7ryh-review-the-poster-child-of-mini-pcs-still-leads-the-pack.html> [últim accés novembre 2017].

[43] Rfsolutions.com.uk. *RF Receiver + Decoder Hybrid* [Internet]. Rfsolutions. Disponible a: <https://www.rfsolutions.co.uk/downloads/1456161817DS-HIRK-433AP.pdf> [últim accés novembre 2017].

[44] RBR Ltd. *XR-420/620 Series. Multichannel Loggers.* [Internet]. Ottawa: Rbr Europe Ltd. Disponible a: <http://www.lunus.com.br/wp-content/uploads/2012/10/Registrador-multi-canais-XR.pdf> [últim accés novembre 2017].

[45] Ni.com (2015). *National Instruments USB 6008/6009. Bus-powered Multifunctions DAQ USB Device* [Internet]. National Instruments. Disponible a: <http://www.ni.com/pdf/manuals/371303n.pdf> [últim accés desembre 2017].

[46] Ni.com. *National Instruments: sistemas de Pruebas, Medidas y Embebidos* [Internet]. Disponible a: <http://www.ni.com> [últim accés desembre 2017].

[47] Ti.com. Texas Instruments. *LM35 Precision Centigrade Temperature Sensors.* [Internet]. Disponible a: <http://www.ti.com/lit/ds/symlink/lm35.pdf> [últim accés desembre 2017].

[48] Es.rs-online.com. *IPS Series Industrial Pressure Sensor.* [Internet]. Disponible a: <http://docs-europe.electrocomponents.com/webdocs/1541/0900766b81541a41.pdf> [últim accés desembre 2017].

## CAPÍTOL 4

[49] Irazabal JM, Blozis S (2003). *AN10216-01: I<sup>2</sup>C Manual* [Internet]. Philips Semiconductors Disponible a: <https://www.nxp.com/docs/en/application-note/AN10216.pdf> [últim accés novembre 2017].

## CAPÍTOL 5

[50] Arduino.cc. [Internet]. Disponible a: <https://www.arduino.cc/> [últim accés novembre 2017].

[51] Orcad.com. *Electronic Hardware and PCB Design software* [Internet]. Disponible a: <http://www.orcad.com/> [últim accés novembre 2017].

[52] Autodesk.com *PCB Design and Schematic Software. EAGLE*. [Internet]. Disponible a: <https://www.autodesk.com/products/eagle/overview> [últim accés desembre 2017].

[53] Kicad-pcb.org. *KiCad EDA* [Internet]. Disponible a: <http://kicad-pcb.org/> [últim accés desembre 2017].

[54] Fritzing.org. *Fritzing* [Internet]. Disponible a: <http://fritzing.org/home/> [últim accés desembre 2017].

[55] Digi.com. *M2M Communications , Remote Monitoring and Management* [Internet]. Digi. Disponible a: <https://www.digi.com/> [últim accés desembre 2017].

[56] Digi International Inc (2017). *XBee®/XBee-PRO S2C Zigbee®. RF Module* [Internet]. Digi. Disponible a: <https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf> [últim accés desembre 2017].

[57] Public Image (2015). *XBee Pinout*. [Internet]. Disponible a: [https://imagemag.ru/img-ba\\_xbee-s2c-pinout.html](https://imagemag.ru/img-ba_xbee-s2c-pinout.html) [últim accés desembre 2017].

[58] Adafruit.com. *Adafruit industries, unique and fun DIY electronics and kits* [Internet]. Adafruit. Disponible a: <https://www.adafruit.com/> [últim accés novembre 2017].

## ALTRES

Draw.io. [Internet]. Disponible a : <https://www.draw.io/> [últim accés desembre 2017].

Voxtechnologies.com. *Industrial PC and Enterprise Networking Source* [Internet]. Disponible a: <http://www.voxtechnologies.com> [últim accés novembre 2017].

Wdlsystems.com. *WDL Systems. The embedded Products Source* [Internet]. Disponible a: <http://www.wdlsystems.com> [últim accés novembre 2017]

Oceanografialitoral.com. *INNOVA Oceanografia Litoral* [Internet]. Disponible a: <http://www.oceanografialitoral.com> [últim accés desembre 2017].

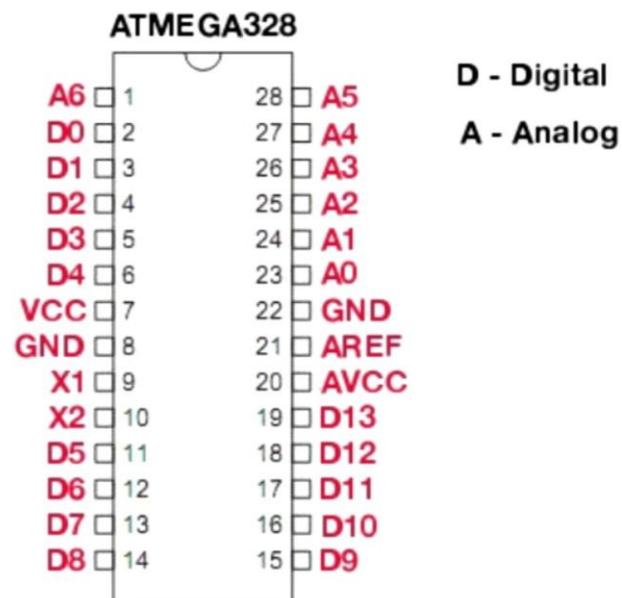
Martínez E, Toma DM, Del Río J, García O, Bghiel I. *ID41-SWE bridge: software interface for plug and work instrument integration into marine observation platforms* [Internet]. InstrumentationViewpoint. Disponible a: <http://upcommons.upc.edu/bitstream/handle/2117/100192/ID41.pdf;jsessionid=7A205B2AA71DBFC0B6ECCD9092F0CCF4?sequence=1> [últim accés novembre 2017].



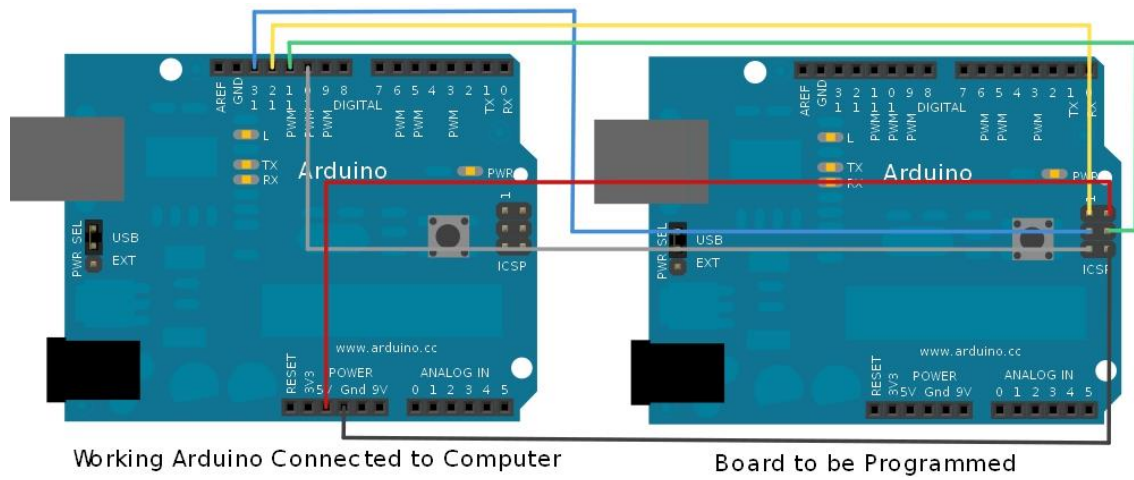
## ANNEX I. IMATGES

- 1. Pins del ATmega328P-PU
- 2. Càrrega del bootloader
- 3. Comparativa entre models de Raspberry Pi
- 4. Esquema Arduino Uno
- 5. Esquemàtic del prototip II (Disseny final del sistema d'encesa) fet amb Eagle
- 6. PCB del prototip II (Disseny final del sistema d'encesa) fet amb Eagle

### 1. Pins del ATmega328P-PU



## 2. Càrrega del bootloader

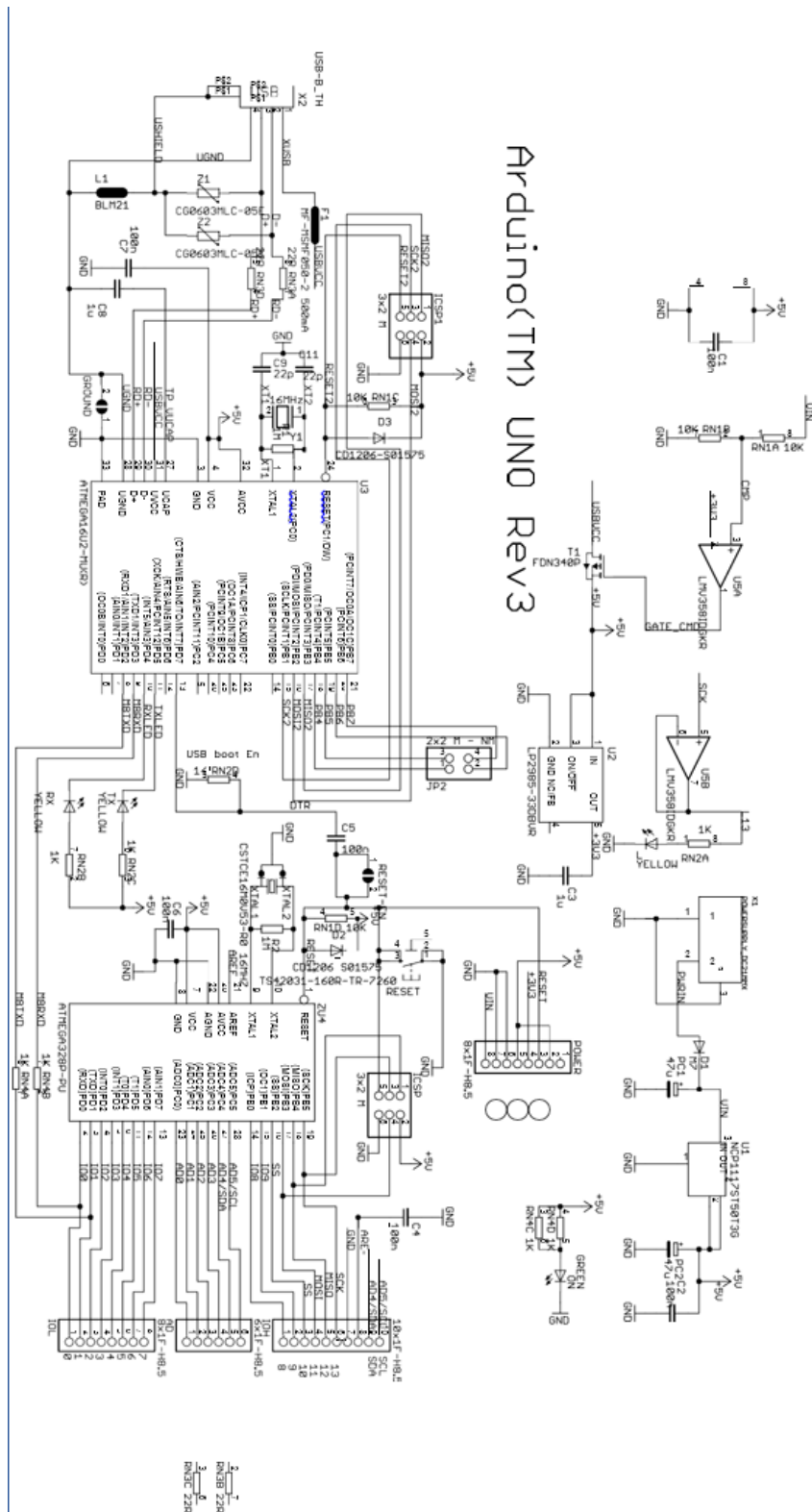


## 3. Comparativa entre models de Raspberry Pi



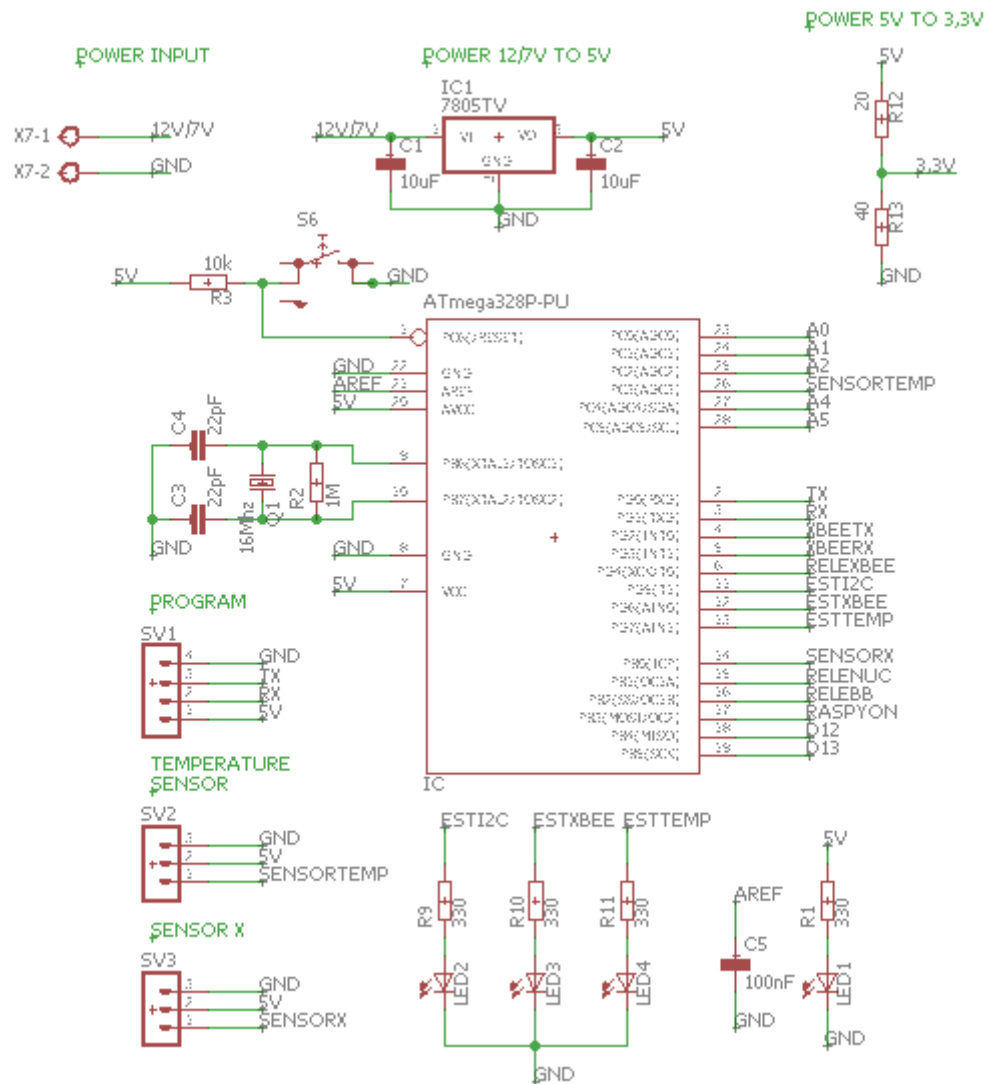


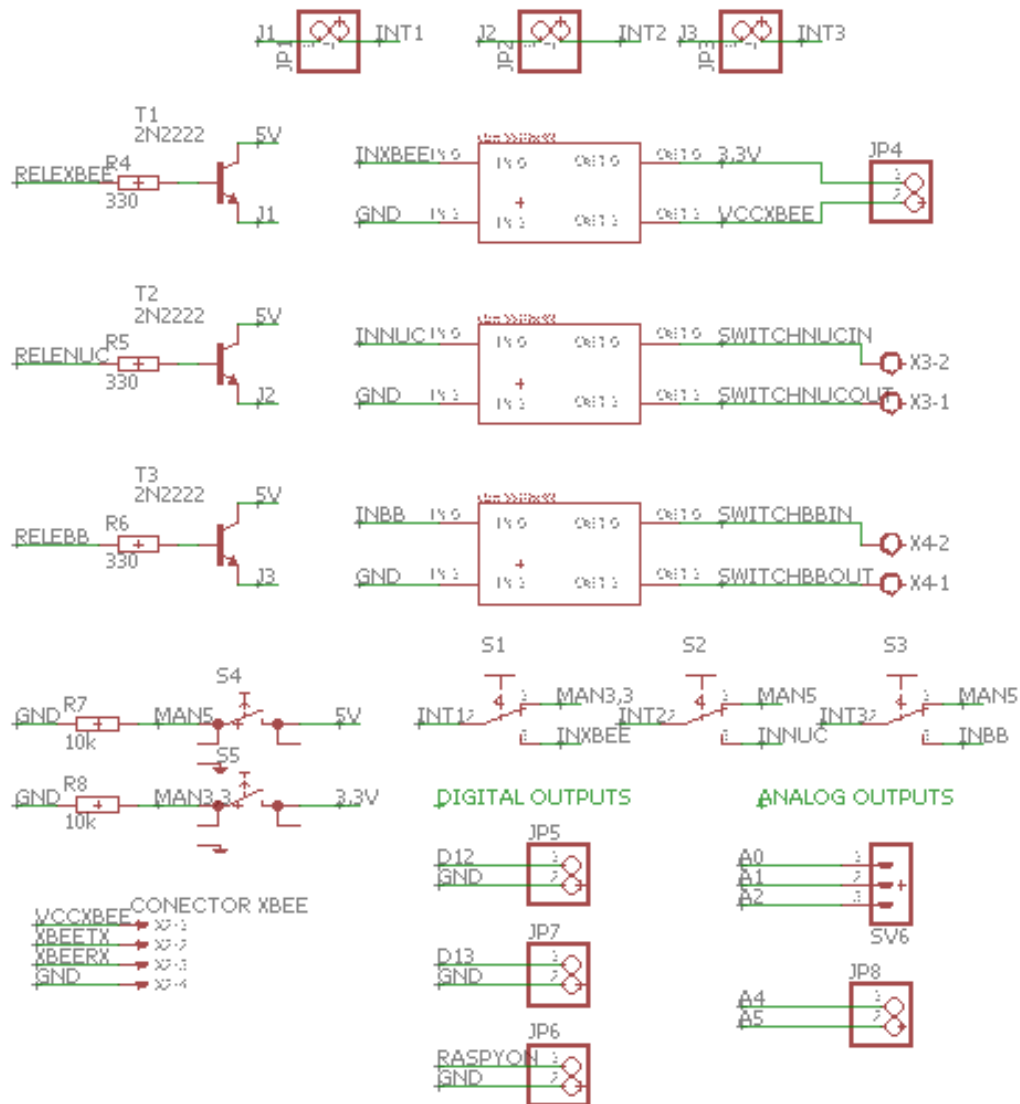
#### 4. Esquema Arduino Uno

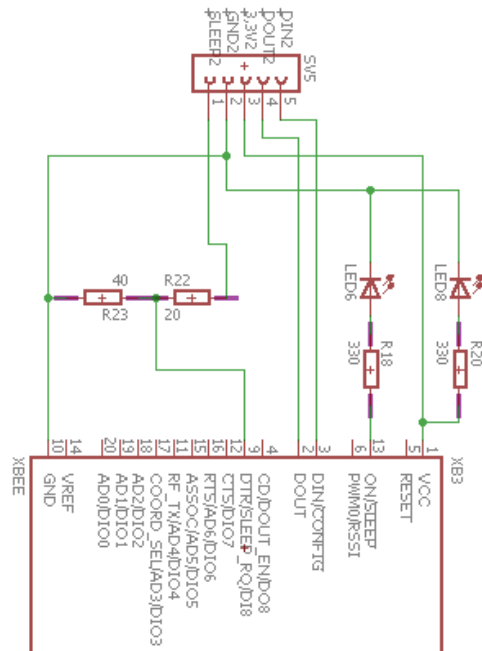
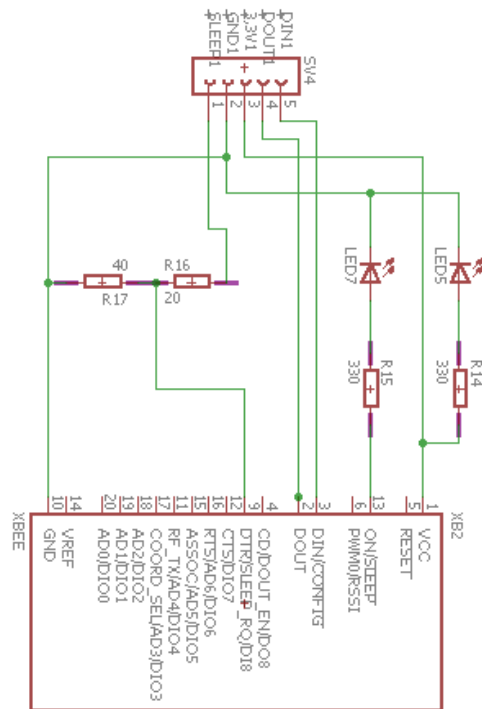


## 5. Esquemàtic del prototip II (Disseny final del sistema d'encesa) fet amb Eagle

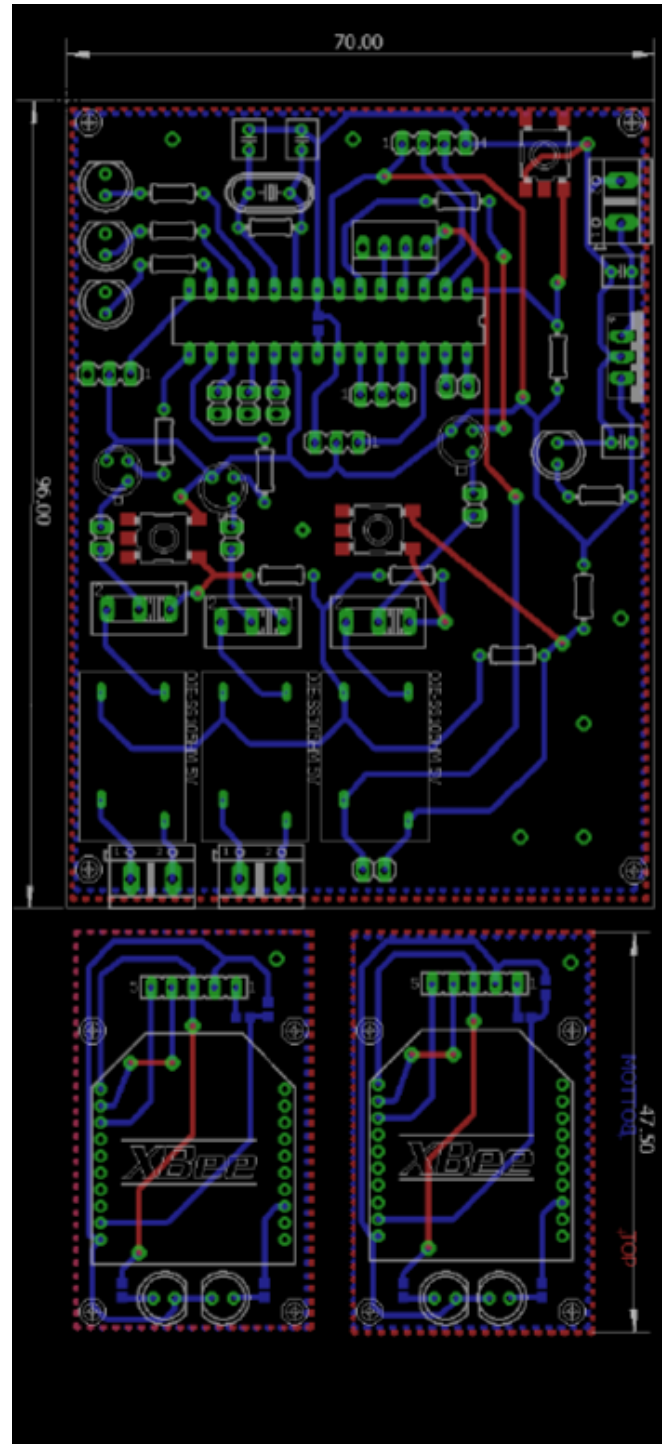
S'adjunta finalment com a pdf dins el CD. També s'inclouen els arxius amb Eagle.







## 6. PCB del prototip II (Disseny final del sistema d'encesa) fet amb Eagle





## **ANNEX 2. PROGRAMACIÓ**

1. Lector voltatge
2. Test encesa
3. Xbee bàsic
4. Lector baromètric RS Components
5. Programa final del sistema d'encesa del Guanay 3.0.

S'inclouen tots els programes a l'interior del cd junt amb el programa fet amb Labview.

### **1. Lector voltatge**

```
void setup() {  
  
  Serial.begin(9600);  
  
  pinMode(A0,OUTPUT);// Lectura pin analógico 0  
  
}  
  
void loop() {  
  
  int a=analogRead(A0); // Variable local 'a'  
  
  Serial.println(a); // Mostrar por el puerto serie la variable  
  
  delay(1000); // Espera de 1000 ms  
  
}  
  
// FIN
```

## **2. Test encesa**

```
//PROTOTIPO I y PROTOTIPO II

void setup() {

  pinMode(13,OUTPUT);

  pinMode(12,OUTPUT);

  pinMode(11,OUTPUT);

  pinMode(10,OUTPUT);

  pinMode(9,OUTPUT);

  pinMode(8,OUTPUT);

  pinMode(7,OUTPUT);

  pinMode(6,OUTPUT);

  pinMode(5,OUTPUT);

  pinMode(4,OUTPUT);

  pinMode(3,OUTPUT);

  pinMode(2,OUTPUT);

  pinMode(1,OUTPUT);

  pinMode(0,OUTPUT);

}

void loop() {

  // para ver todos los pines apagados

  digitalWrite(13,LOW);
```



```
digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

digitalWrite(7,LOW);

digitalWrite(6,LOW);

digitalWrite(5,LOW);

digitalWrite(4,LOW);

digitalWrite(3,LOW);

digitalWrite(2,LOW);

digitalWrite(1,LOW);

digitalWrite(0,LOW);

//o bien para ver todos los pines encendidos.

/*digitalWrite(13,HIGH);

digitalWrite(12,HIGH);

digitalWrite(11,HIGH);

digitalWrite(10,HIGH);

digitalWrite(9,HIGH);

digitalWrite(8,HIGH);

digitalWrite(7,HIGH);

digitalWrite(6,HIGH);
```

```
digitalWrite(5,HIGH);

digitalWrite(4,HIGH);

digitalWrite(3,HIGH);

digitalWrite(2,HIGH);

digitalWrite(1,HIGH);

digitalWrite(0,HIGH);*/

}

// FIN
```

### **3. Xbee bàsic**

```
#include <SoftwareSerial.h>

SoftwareSerial xbee(0, 1); // o bien SoftwareSerial xbee(2, 3); // RX, TX

char on = 'i'; // Carácter para encender el led

char off = 'o'; // Carácter para apagar el led

int lectura = 0; //Variable lectura

int led = 8; // Led en el pin digital 8

void setup() {

    Serial.begin(9600); // Baud rate monitor serie

    Serial.println( "XBees connected" );

    pinMode(led, OUTPUT); // Led como salida

    xbee.begin(9600); // Baud rate Xbee

}
```

```
void loop() {  
  
    lectura = xbee.read(); // Lectura del xbee  
  
    if( lectura == on){  
  
        digitalWrite(led, HIGH); // Encender el led  
  
        Serial.println( lectura ); // Imprimir lectura en el monitor serial  
  
        Serial.println( "LED ON" ); // Imprimir este mensaje  
  
    }  
  
    if(lectura == off)  
  
    {  
  
        digitalWrite(led, LOW); // Apagar el led  
  
        Serial.println( lectura ); // Imprimir en el monitor serie  
  
        Serial.println( "LED OFF" ); // Imprimir este mensaje  
  
    }  
  
    delay(100); // Espera 100 ms  
  
}  
  
// FIN
```

#### **4. Lector baromètric RS Components.**

/\*

\* IPS Series Industrial Pressure Sensor

\* 797-5046 RS; <https://es.rs-online.com/web/p/sensores-de-presion/7975046/> ; 160,88 €

\* Serial No: 403643

\* OUTPUT: 0 to 6 Bar G

\* OUTPUT: 0 to 5 Vdc

\*

\* Development by Gerard Bonavida

\* gerbon.13@gmail.com

\*

\* Wire:

\* Red: Vcc=9-32 Vdc (11Vdc in my ideal case)

\* Black: GND

\* Yellow: Signal - (I use this wire, only add 100000.0)

\* Green: Signal +

\* As minus voltage, measure have more error.

\*

\*/

float a;//sensor value to arduino uno

void setup() {

Serial.begin(9600);

```
pinMode(0,INPUT);//use Analog 0 as input

}

void loop() {

  a=analogRead(0);//read analog0 and save in the variable "a".

  float voltage= (5.0*a)/1024.0;//conversion factor to pas as voltage

  float pascals= (600000.0*voltage/5.0)+100000;

  Serial.print(voltage);

  Serial.println(" V");

  Serial.print(pascals);

  Serial.println(" Pa");

  delay(1000);

}

//FIN
```

## 5. Programa final del sistema d'encesa del Guanay 3.0

```
/*  
  
* Designed by Gerard Bonavida Vila  
  
* gerbon.13@gmail.com  
  
*  
  
* Guanay 3.0 - Boot System  
  
* 3 relay and raspy pin  
  
*/  
  
#include <Wire.h>  
  
#include <Adafruit_BMP085.h>  
  
#include <SoftwareSerial.h>  
  
  
Adafruit_BMP085 bmp;  
  
SoftwareSerial xbee(2,3); //en la PCB es 2,3!!  
  
  
//XBEE variables//  
  
char on = 'i';//de encender componentes  
  
char off = 'o';//de cerrar componentes  
  
char progon= 'p';//de program  
  
char progoff= 'e';//de escape// no se utiliza finalmente.  
  
char reinixbee= 'r';//de reinicio comunicacion xbee  
  
char temp= 't';//pedir la temperatura del sensor LM35 en A5
```

```
int lectura= 0;//variable inicio

bool varxbee=1;//variable on

bool varybee=0;//variable off

bool varzbee=1;//variable program on

bool varwbee=0;//variable for "program off". Only switch on raspy

//y despues si eso se apaga o se enciende la raspy


//LDR variables//

String Str1= "en superficie";//variable global

String Str2= "hundido";//variable global

String salidaprint;//salida global


/*PINS

* Digital:

* 0->rx ; 1->tx ; 2->rx xbee ; 3->tx xbee ; 4-> relexbee; 5-> esti2c(led vermell) ; 6-> estxbee(led verd)
; 7-> esttemp(led amar)

* 8-> sensorx (¿digital?) ; 9-> relenuc ; 10-> releBB; 11->raspyon; 12&13-> - ;

* Analog:

* A0,A1,A2,A4,A5-> - ; A3-> Sensor Temperatura/LDR;

*/

void setup() {
```

```
Serial.begin(9600);//serial port baud rate

Serial.println("System by Gerard Bonavida");

xbee.begin(9600);//xbee baud rate

Serial.println("Xbee on");

//PINES ACTUALES

pinMode(4,OUTPUT);//rele xbee

digitalWrite(4,HIGH);//activacion rele siempre activo (consume pero es demo)

pinMode(5,OUTPUT);//led rojo: 1)estado I2C. 2)estado rele nuc 3)hundido

pinMode(6,OUTPUT);//led verde: 1)estado xbee. 2)en superficie

pinMode(7,OUTPUT);//led amarillo: 1)estado temperatura 2)estado rele beagle bone

pinMode(8,INPUT);//

pinMode(9,OUTPUT);//rele nuc

pinMode(10,OUTPUT);//rele beagle bone

pinMode(11,OUTPUT);//raspyon (pin para encender raspberry)

pinMode(12,OUTPUT);//led exclusivo para ver si hay transmision raspyon

pinMode(A3,OUTPUT);//sensor temperatura/LDR

//pinMode(7,OUTPUT);//led blanco prueba rele nuc

//pinMode(8,OUTPUT);//led rojo prueba rele beagle bone

//pinMode(9,OUTPUT);//led amarillo prueba rele raspberry

//pinMode(11,OUTPUT);//led hundido

//pinMode(12,OUTPUT);//led en superficie

//pinMode(13,OUTPUT);
```



```
pinMode(A0,OUTPUT);//lector LDR

pinMode(A5,OUTPUT);//LM35

/*if (!bmp.begin()) {

Serial.println("sensor BMP085 no encontrado!");

while (1) {}

}*/

}

void loop() {

    barometro();

    leds();//desactivada

    ldr();

    LM35();

    //transmisioni2c(); //future transmissi3n with other MCU

    delay(100);

    lectura=xbee.read();

    digitalWrite(6,HIGH);//led verde ON programa run

    //delay(100);

    //digitalWrite(6,LOW);//led verde OFF program run

    if(lectura==on && varxbee==1)//dos opciones, que se encienda desde 0 o desde la programacion
on

    {systemon();} //se puede tambien configurar, lo unico que no se cerrara nunca la raspy

    else if(lectura==off && varybee==1)

    {systemoff();}
```

```
else if(lectura==progon && varzbee==1)//
{systemprogramon();}

else if(lectura==reinixbee)
{systemreinixbee();}

else{}

}

//start modo encendido

void systemon(){

    if(varwbee==0){

        digitalWrite(11,HIGH);//flanco subida raspy on=ON

        digitalWrite(12,HIGH);//led raspyon ON

        delay(250);

        digitalWrite(11,LOW);//flanco bajada raspy

        digitalWrite(12,LOW);//led raspyon OFF

        delay(250);

        digitalWrite(10,HIGH);//flanco subida rele bb=ON

        digitalWrite(7,HIGH);//led rele bb on

        delay(250);

        digitalWrite(10,LOW);//flanco bajada rele bb

        digitalWrite(7,LOW);//led rele bb off

        delay(250);

        digitalWrite(9,HIGH);//flanco subida rele nuc=ON
```

```
digitalWrite(5,HIGH);//led rojo estado nuc ON

delay(250);

digitalWrite(9,LOW);//flanco bajada rele nuc

digitalWrite(5,LOW);//led rojo estado nuc OFF

delay(2000);//para evitar que pulses muchas veces o errores

varxbee=0;

varybee=1;

varzbee=0;

varwbee=0;

}

else{//w=1 for ex. Only switch on raspy.

digitalWrite(9,LOW);//nuc rele low

digitalWrite(5,LOW);//led rojo estado nuc OFF

digitalWrite(10,LOW); //bb rele low

digitalWrite(7,LOW);//led amarillo estado bb OFF

digitalWrite(11,HIGH);//flanco subida rele raspy=ON

digitalWrite(12,HIGH);//led raspyon ON

delay(250);

digitalWrite(11,LOW);//flanco bajada rele raspy

digitalWrite(12,LOW);//led raspyon OFF

delay(2000);//para evitar que pulses muchas veces o errores

varxbee=0;
```

```
    varybee=1;

    varzbee=0;

    varwbee=0;

}

}

void systemoff(){

    if(varwbee==0){

        digitalWrite(9,HIGH);//flanco subida rele nuc=ON

        digitalWrite(5,HIGH);//led rojo estado nuc ON

        delay(250);

        digitalWrite(9,LOW);//flanco bajada rele nuc

        digitalWrite(5,LOW);//led rojo estado nuc OFF

        delay(250);

        digitalWrite(10,HIGH);//flanco subida rele bb=ON

        digitalWrite(7,HIGH);//led amarillo estado bb ON

        delay(250);

        digitalWrite(10,LOW);//flanco bajada rele bb

        digitalWrite(7,LOW);//led amarillo estado bb OFF

        delay(250);

        //estas 3 lineas no serviran para nada ya que la raspy no se apaga!!! sistema seguridad

        //digitalWrite(11,HIGH);//flanco subida rele raspy=ON

        //delay(250);
```

```
//digitalWrite(11,LOW);//flanco bajada rele raspy  
  
varxbee=1;  
  
varybee=0;  
  
varzbee=1;  
  
varwbee=0;  
  
}  
  
else{//varwbee=1  
  
digitalWrite(11,LOW);//flanco bajada rele raspy  
  
digitalWrite(12,LOW);//led raspyon OFF  
  
digitalWrite(9,HIGH);//flanco subida rele nuc=ON  
  
digitalWrite(5,HIGH);//led rojo estado nuc ON  
  
delay(250);  
  
digitalWrite(9,LOW);//flanco bajada rele nuc  
  
digitalWrite(5,LOW);//led rojo estado nuc OFF  
  
delay(250);  
  
digitalWrite(10,HIGH);//flanco subida rele bb=ON  
  
digitalWrite(7,HIGH);//led amarillo estado bb ON  
  
delay(250);  
  
digitalWrite(11,LOW);//flanco bajada rele bb  
  
digitalWrite(7,LOW);//led amarillo estado bb OFF  
  
delay(250);  
  
varxbee=1;
```

```
    varybee=0;

    varzbee=1;

    varwbee=0;

}

}

void systemprogramon(){

    digitalWrite(11,LOW);//relé para activar raspberry apagado

    digitalWrite(12,LOW);//led raspyon OFF

    digitalWrite(10,HIGH);//flanco subida rele bb=ON

    digitalWrite(7,HIGH);//led amarillo estado bb ON

    delay(250);

    digitalWrite(10,LOW);//flanco bajada rele bb

    digitalWrite(7,LOW);//led amarillo estado bb OFF

    delay(250);

    digitalWrite(9,HIGH);//flanco subida rele nuc=ON

    digitalWrite(5,HIGH);//led rojo estado nuc ON

    delay(250);

    digitalWrite(9,LOW);//flanco bajada rele nuc

    digitalWrite(5,LOW);//led rojo estado nuc OFF

    delay(250);

    varxbee=1;

    varybee=1;
```

```
varzbee=0;

varwbee=1;

}

void systemreinixbee(){//modo para reiniciar el xbee

    delay(500);

    digitalWrite(4,LOW);//rele nuc OFF

    digitalWrite(6,LOW);//led verde estado xbee OFF

    delay(500);

    digitalWrite(4,HIGH);//rele nuc ON

    digitalWrite(6,HIGH);//led verde estado xbee ON

    delay(2000);

}

//end modo encendido

void barometro(){//BMP085 sensor to 3.3V (NOT 5.0V!)

    /*Serial.print("Temperature = ");

    Serial.print(bmp.readTemperature());

    Serial.println(" *C");

    //NO VA

    if(bmp.readTemperature(>45){//if the temperature is superior to 45 degrees...buzzer on.

        digitalWrite(0,HIGH);

        //digitalWrite(PINBUZZER,HIGH);
```

```
Serial.println("ALTA TEMPERATURA!!");

else{}

//N.V.,;

Serial.print("Pressure = ");

Serial.print(bmp.readPressure());

Serial.println(" Pa");

// Calculate altitude assuming 'standard' barometric

// pressure of 1013.25 millibar = 101325 Pascal

Serial.print("Altitude = ");

Serial.print(bmp.readAltitude());

Serial.println(" meters");

Serial.print("Pressure at sealevel (calculated) = ");

Serial.print(bmp.readSealevelPressure());

Serial.println(" Pa");

// you can get a more precise measurement of altitude

// if you know the current sea level pressure which will

// vary with weather and such. If it is 1015 millibars

// that is equal to 101500 Pascals.
```



```
Serial.print("Real altitude = ");

Serial.print(bmp.readAltitude(101500));

Serial.println(" meters");


digitalWrite(13,HIGH);//led de comprovacion de I2C

delay(100);

digitalWrite(13,LOW);

//Serial.println();

//delay(500);*/

}

void leds(){//esta desactivada

/*digitalWrite(13,HIGH);

delay(1000);

digitalWrite(13,LOW);

digitalWrite(12,HIGH);

delay(1000);

digitalWrite(12,LOW);

digitalWrite(11,HIGH);

delay(1000);

digitalWrite(11,LOW);*/

}

void LM35(){
```

```
if(lectura==temp){

    int temperatura = analogRead(A5);

    float milivolts = (temperatura/ 1023.0) * 5000;

    float graus = milivolts / 10;

    Serial.print(graus);

    xbee.write(graus);

    Serial.println(" C");

    xbee.write(" C");

    delay(1000);

}

else {}

}

void ldr(){

    //inicio variables

    int a;

    int amax;

    int amin;

    int histeresis=10;

    int oscuridad=500;//valor de la oscuridad//10k=500 100K=30

    bool lastsalida=0;

    bool salida;

    //String Str1= "en superficie";//variable global
```

```
//String Str2= "hundido";//variable global
```

```
//String salidaprint;//salida global
```

```
//inicializa lector
```

```
pinMode(A0,INPUT);
```

```
a=analogRead(A0);
```

```
amax=oscuridad+histeresis;
```

```
amin=oscuridad-histeresis;
```

```
Serial.println(a);
```

```
if(a<amin && a<amax){
```

```
    salida=0;//en superficie
```

```
    salidaprint=Str1;
```

```
    /*Serial.println(salidaprint);
```

```
    digitalWrite(11,LOW);
```

```
    digitalWrite(12,HIGH);
```

```
    digitalWrite(13,LOW);*/
```

```
    lastsalida=salida;
```

```
}
```

```
if(a>=amax && a>amin){
```

```
    salida=1;
```

```
    salidaprint=Str2;
```

```
    Serial.println(Str2);
```

```
    /*digitalWrite(11,HIGH);
```

```
digitalWrite(12,LOW);

digitalWrite(13,LOW);*/

lastsalida=salida;

}

if(a>=amin && a<=amax){

    salida=lastsalida;

    salidaprint=salidaprint;

    Serial.println(salidaprint);

    salida=lastsalida;

}

//delay(2000);

/*digitalWrite(11,LOW);

digitalWrite(12,LOW);

digitalWrite(13,LOW);

delay(2000);*/

}

// FIN
```



